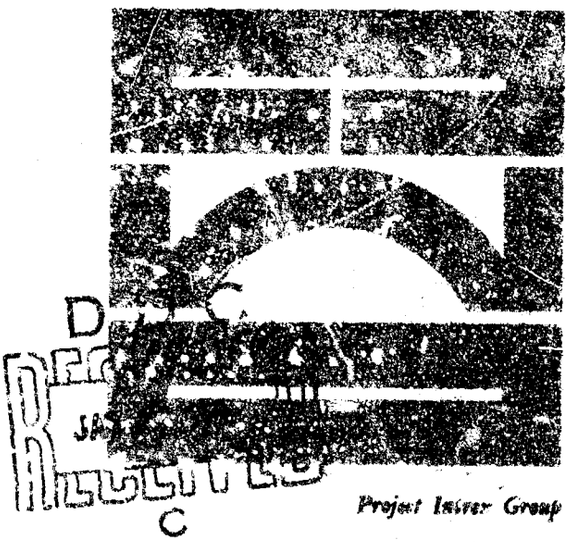
July, 1968

REPORT ESL-R-345
M.I.T. PROJECT DSR 70054
Research Grant NSG-472 (Part)

RAPID PRE-INDEXING BY MACHINE

William R. Kampe II



Electronic Systems Laboratory — *Project Intrex Group*

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

Department of Electrical Engineering

70

July, 1968 Report ESL-R-355 Copy 38

RAPID PRE-INDEXING BY MACHINE

by

William R. Kampe II

The work described in this document was performed as part of Project Intrex under Research Grant NSFC-472 (Part) awarded to the Massachusetts Institute of Technology by the National Science Foundation and the Advanced Research Projects Agency of the Department of Defense. This grant is designated as M.I.T. DSR Project No. 70054.

Electronic Systems Laboratory
Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

BIBLIOGRAPHIC DATA SHEET	1. Report No. ESL-R-355	2.	3. Recipient's Accession No.
4. Title and Subtitle: Rapid Pre-Indexing by Machine			5. Report Date: July, 1968 6.
7. Author(s): William R. Kampe II			8. Performing Organization Rept. No. ESL-R-355
9. Performing Organization Name and Address: Electronic Systems Laboratory, Massachusetts Institute of Technology, Bldg. 35-311, Cambridge, MA 02139			10. Project/Task/Work Unit No. 11. Contract/Grant No. NSFC-472(Part)
12. Sponsoring Organization Name and Address: National Science Foundation, Office of Science Information Service, 1800 G. Street, N. W., Washington, D. C. 20550			13. Type of Report & Period Covered: Technical Report 14.

15. Supplementary Notes

16. Abstracts: This report describes the development of a new method of subject indexing by machine for documents in the Project INTREX catalog. The purpose of the system is to allow new documents to be placed online quickly in the computer-stored Intrex catalog. The system that is developed makes use of human-generated subject terms of existing Intrex documents as a basis for generating index terms for new documents. The pre-indexing system operates on only the title and abstract of a document in generating a pre-index for the document. The analysis of documents already containing human-generated subject indexes consisted of comparing the titles and abstracts of the documents to their subject indexes. A large dictionary with data about word usage was obtained from these comparisons. The dictionary served as a guide for the later pre-indexing of new documents. Three variations of the automatic pre-indexing method have been developed, tested, and evaluated. Two methods show promise for operational use in the Intrex system.

17. Key Words and Document Analysis. 17a. Descriptors

Subject indexing
Automatic indexing
Information retrieval
Words (language)

17b. Identifiers/Open-Ended Terms

Project Intrex
Automatic pre-indexing
Free indexing
Word frequency

17c. COSATI Field/Group 5B

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Release unlimited	UNCLASSIFIED	68
	20. Security Class (This Page) UNCLASSIFIED	22. Price

FOREWORD

Except for minor editorial changes, this report is the thesis submitted by Mr. William R. Kampe II to the Electrical Engineering Department, Massachusetts Institute of Technology, in partial fulfillment of the requirements for the degree of Master of Science. A few alterations in the original wording have been made throughout the text in an effort to enhance clarity, and several pages have been reformatted; otherwise the manuscript remains as submitted.

J. F. Reintjes

Professor of Electrical Engineering

CONTENTS

CONTENTS (Contd.)

LIST OF FIGURES

ACKNOWLEDGMENT

I am greatly indebted to Professor J. Francis Reintjes, thesis supervisor, for his guidance, suggestions, and critical evaluation of all stages of this research. I thank Peter Kugel for his continued encouragement and for his endless patience in discussing new technical approaches. Thanks are also expressed to Messrs. Robert Kusik, Richard Marcus, and Alan Benenfeld for their explanations of the Project INTREX cataloging process.

In addition, I am grateful to Professor Lynwood Bryant, who spent much time studying the manuscript and making pertinent suggestions for improvement. I also thank the Publications and Drafting Personnel of the Electronic Systems Laboratory for their efforts in typing and assembling the final document.

This research was supported by the Department of Electrical Engineering and by Project INTREX. Project INTREX is supported through grants from the Carnegie Corporation and the Council on Library Resources, Inc., and through Contract NSFC-472 from the National Science Foundation and Advanced Research Projects Agency.

CHAPTER I

THE NEED FOR PRE-INDEXING

This research involves an investigation of the feasibility of using automatic machine-generated subject indexes as a possible means of expediting the cataloging process for new documents to be added to the Project INTREX library. An online library in a fast-changing technical field needs a method of putting information about a new document into the library system quickly. A possible method is to generate an automatic pre-index from only the title and abstract of a new document. The pre-index is intended to serve as a temporary subject index until human catalogers can replace the pre-index with a full subject index based on an examination of the entire new document.

The pre-indexing schemes investigated are based on past experience gained from a computer analysis of the human-generated indexes of a set of documents. The experience thus obtained through comparison of title and abstracts to corresponding subject indexes is used as a guide for generating, by machine, a pre-index for a new document to be added to the system. The pre-indexing scheme being proposed differs from previous work in automatic indexing in that the pre-indexing scheme makes extensive use of prior human indexing in an automatic fashion.

The INTREX Environment

Project INTREX (Information Transfer Experiments) seeks to exploit the multiaccess computer operated in an online mode as a basis for a machine-stored library. The INTREX collection will consist of approximately 10,000 documents in selected fields of materials science and engineering. Each document will be cataloged in depth and the catalog will be stored in a multiaccess computer. Full text of each document will be stored on microfilm and made accessible under computer control. A user of the library will be able to search the catalog by means of a time-shared computer system. Special remote consoles will allow him to obtain displays of the full text of documents, or hard copies if he so desires.

At present, INTREX is still in the developmental stage. There are no users of the system, as yet. Nevertheless, over 500 documents have been fully cataloged and placed into the computer data base, accessible to the INTREX programmers on a time-sharing computer system. It is this data base which is used for this research effort. The INTREX programmers have written numerous programs to facilitate access to the data base.

The cataloging record for each document of the collection comprises many of the standard items found on the typical catalog cards of conventional libraries. The title and author, publication date, publisher, and number of pages are included. In addition, an INTREX record for a document includes information such as the language in which the document is written, the abstract, and a subject index. Each particular type of information is placed in its own field (title, author, publisher, catalog number, and so forth) in a document record. Most fields in a document record can be obtained in a purely routine manner. This information comes with the document and needs only to be transferred to the document record. The abstract, when included in the record, also arrives with the document and has been written by the author or an abstracting agency. Nearly all documents of the INTREX collection include an abstract. If not, the cataloger substitutes an excerpt.

In a document record, the one field which requires analysis of the document in order to be generated is the subject index. Since the subject index is of prime interest in pre-indexing, it will be examined here in some detail. Of 49 possible descriptive fields being used by INTREX for a document, the process of generating the subject index field alone consumes over half of an indexer's time even for short journal articles. For longer documents, the subject index takes a larger part of the total time. The purpose of pre-indexing by machine is to relieve the indexers of the pressing nature of the burden of writing the subject index for the document.

In generating the subject index for a document, the indexers may draw upon the material of the full document, including the title, abstract, and text. The subject index comprises several subject terms which describe the content of the document. Each subject term may range in length from a single word to one or more noun phrases containing

several words. To every subject term, the indexer assigns a relevancy weight, which indicates how the term is used and how important it is in the document. This weighting system is a key feature of Project INTREX, since it is hoped that the weights will allow more accurate retrieval of pertinent documents.

A subject term may be labeled with any of five weights, numbered from 0 through 4. The numbers 0 and 4 have special significance, whereas the numbers 1 through 3 indicate specific subject matter of the document. A weight of 1 indicates that the subject term describes the primary topic of the document; weight (2) designates a secondary topic; and weight (3), a much less central topic. The special weight of 4 is assigned to terms representing mathematical tools, instrumental tools, or applications which are cited in the document but which are not central to it. The weight (0) designates a generic class to which the subject matter of a document belongs.

Knowledge of the indexing behavior of the INTREX catalogers provides support for the concept of pre-indexing. The catalogers for Project INTREX are trained and competent in matters of library science. Nevertheless, they are not trained as subject experts in the area of materials science, which is the area of the INTREX document collection. This fact may influence the subject indexes which the indexers generate. Although in writing a subject index, the indexer may use any word that she desires in describing the document, the majority of words used will be those employed by the author of the document. It is natural that an indexer who is not a subject expert should adopt the language of the author in indexing. Since the title and abstract of a document usually provide an excellent indication of the subject matter of that document, a large number of the subject terms of a subject index are borrowed directly from the title and abstract of the document.

Thus, the indexers perform derivative indexing. In pure derivative indexing, the words selected to portray the subject matter of a document are only those words actually used by the author. Salton at Cornell University* has experimented with automatic derivative indexing from

---

*Salton, G. (ed), "Information Storage and Retrieval," Scientific Report No. ISR-11, Department of Computer Science, Cornell University, Ithaca, New York, June, 1966.

title and abstract only. His results are very encouraging to the notion of pre-indexing. Salton found that automatic derivative indexing from the title and abstract only is very nearly as good as automatic indexing from full text.

Motivation for Pre-Indexing

Two factors combine to indicate the possibility of pre-indexing. One factor indicates that pre-indexing is desirable; the other factor, that it is feasible. First, the indexers spend a large part of their time in writing the subject index. It would be desirable if this process could be bypassed, at least temporarily, in order to make the document available to INTREX users quickly. Second, Salton's work and an examination of typical subject indexes reveals that pre-indexing is feasible. Since the title and abstract will be placed with the document index into the computer data base, an automatic method for selecting pertinent words or phrases for a document could generate a pre-index quickly for the document before the subject indexing is completed.

Plan of the Research

Basically, the investigation of pre-indexing divides into two rather distinct phases, as Fig. 1 shows. In Phase I, a number of document

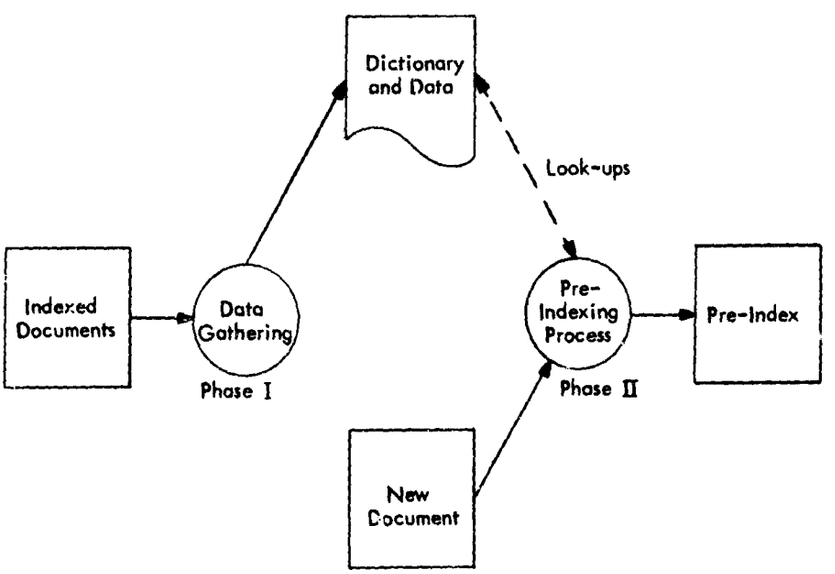


Fig. 1 The Pre-Indexing System

records is automatically analyzed by computer in order to build a dictionary and data base for later use in actual pre-indexing. Further, the analysis is intended to provide clues for designing the pre-indexing methods to be used in Phase II. In Phase II the results of the analysis are applied to the automatic pre-indexing of new documents. Also in Phase II, the pre-indexes that are generated are evaluated to obtain a measure of pre-indexing quality.

CHAPTER II

PHASE I - DATA GATHERING

To analyze profitably the titles, abstracts, and subject indexes of a large number of documents in order to gain experience and data for later pre-indexing, it is necessary first to define the goals of pre-indexing.

The goal of automatic pre-indexing is to extract from the title and abstract of a document the set of words that best describes the contents of the document. In this research, it is assumed that those words in the title and abstract of a document that also appear in the human-generated subject index for the document are the best possible choices for the words of the pre-index. Thus, the goal of pre-indexing is to select from the title and abstract all those words which the human indexers will eventually include in the subject index.

Of course, the pre-index may not always achieve the goal. Figure 2 shows the relationship of the title-and-abstract words to the words of the subject index of a document. The target set of words for a pre-index PI is the set of words TA $\cap$ SI (read: the set of words common to Title/Abstract and Subject Index). Words which are eventually included in the human-generated subject index but are not found in the title and abstract of the document are ignored in this study (the set of words SI-SI $\cap$ TA).

Two types of errors are possible when words are selected for a pre-index. The first is the omission of some words that should be included. As a result of such omissions, the pre-index is incomplete. The second type of error is the inclusion in the pre-index words that should not be included. This second type of error decreases the relevance of the words in the pre-index. Definitions for two measures of pre-index quality which indicate how well the two errors are avoided can now be formulated as follows (see Fig. 3):

> Completeness - The percentage of words in TA $\cap$ SI, the target pre-index, that are actually included in the pre-index
>
> Relevance - The percentage of words in PI, the actual pre-index, that are also included in the target set, TA $\cap$ SI

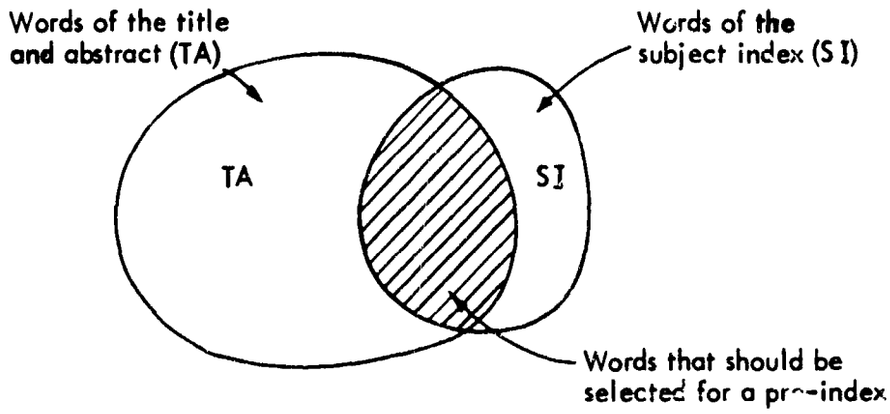


Fig. 2 The Target Set of Words for Pre-Indexing (Shaded Area)

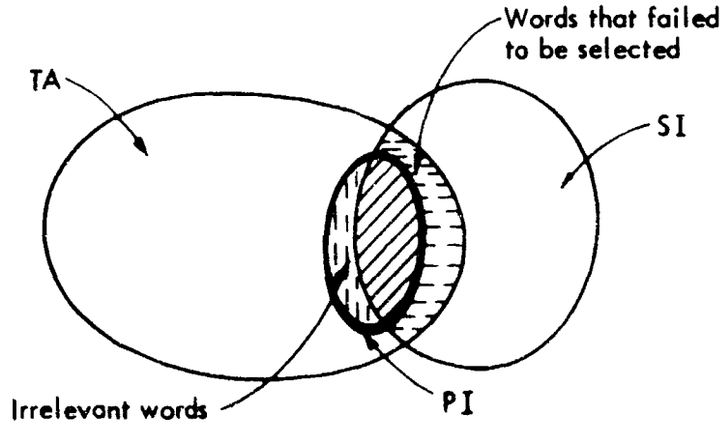


Fig. 3 The Pre-Index

If both completeness and relevance of a pre-index are 100 percent, then the pre-index exactly matches the target set of words, TA ∩ SI.

In this research the standards defined above are based on assumptions that must be remembered when trying to determine the true value of an automatic pre-index. First, the measurements use the human-generated index as reference standards of completeness and relevance. Second, the standards assume that enough information is contained in the title and abstract of a document to extract a valid pre-index from only those two sources. Unfortunately, the quality of the human indexing cannot be evaluated objectively until the INTREX catalog is used operationally. In the meantime it appears safe to assume that the human index is a valid index and reference standard. An examination of subject indexes for several documents shows that most of the words of the subject index appear in the title and abstract of a document. Furthermore, the work of Salton at Cornell indicates that use of only the title and abstract of a document is a good procedure for generating a derivative index by machine. There will be occasion to question the validity of the two measures of pre-indexing later, but for the time being they appear to provide a reasonable basis for an investigation of pre-indexing methods.

Now what remains to be determined is a statistical basis for a pre-indexing system. The next section describes the analysis of many documents which already have a subject index, as well as a title and abstract, so that some decision rules may be devised for the selection of words for a pre-index.

The Analysis of Documents Already Having Subject Indexes

The purpose of analyzing document records which already contain subject indexes is twofold. First, the analysis should yield information for the formulation of possible decision rules for later auto-pre-indexing, and second, the analysis will result in a dictionary of all words that have appeared in the titles and abstracts of all the documents analyzed. Such a dictionary is employed in the pre-indexing of new documents.

In this research, 80 documents already having subject indexes, as well as titles and abstracts, were analyzed by computer according to the techniques formulated below. The subject index of each document

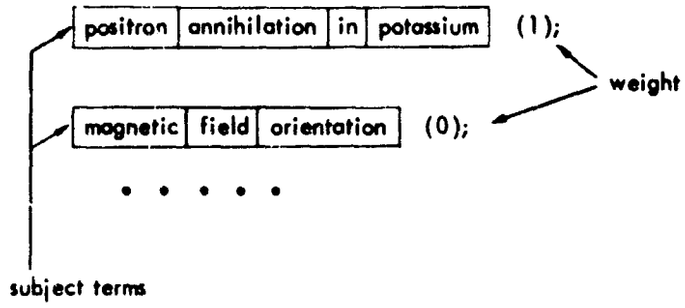


Fig. 4 Typical Subject Index

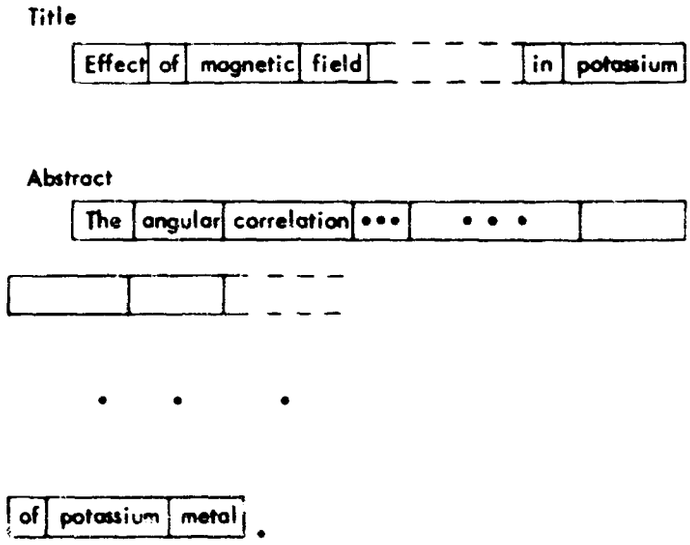


Fig. 5 Typical Title and Abstract

comprises a list of subject terms. A term may include one word or several words combined into a grammatical phrase. A representation of a typical subject index is shown in Fig. 4, and representations of the title and abstract are given in Fig. 5. In order to compare the words in the title and abstract with the words in the subject index, the analysis system in the Phase-I part of the pre-indexing process must convert title and abstract into one list of words and the subject index into another. The converted lists are shown in Fig. 6. The list for

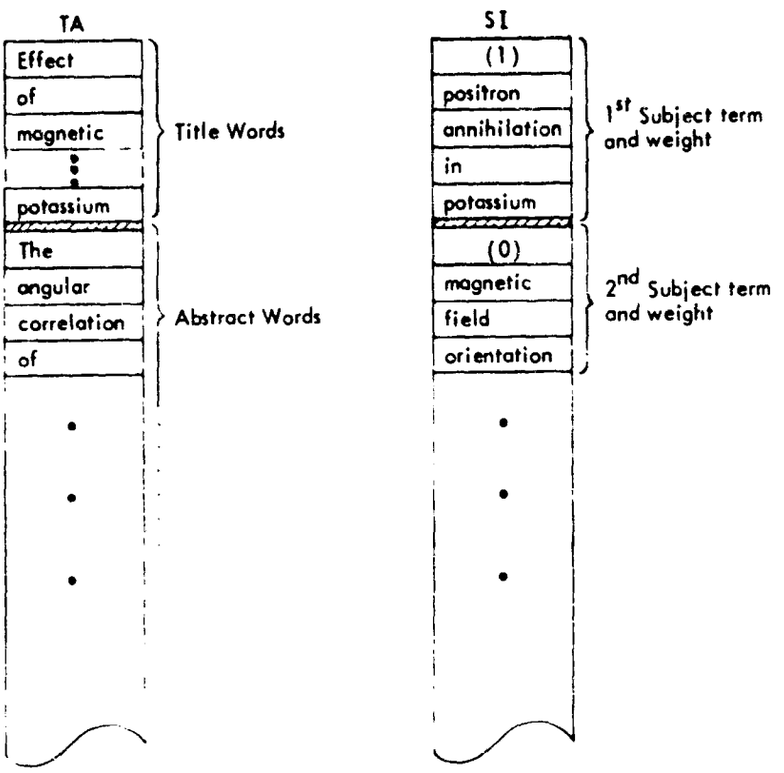


Fig. 6 TA List and SI List

the title and abstract words is called TA; the list for subject index words, SI. The purpose of these lists is to allow simple counts of the usage of words in the title, abstract, and subject index.

In both the title and the abstract sections of TA, each different word is entered only once, regardless of the number of tokens of that word that actually occur in, for example, the abstract of a document. For instance, the word "of" may occur four times in the abstract of a

document. However, "of" will appear only once in the abstract section of TA. If the word "of" also occurs in the title of the document, then "of" will also be entered once in the title section of TA. In SI, each subject term is considered separately. A subject term is reduced to a list of words in a manner similar to that for the listing of the title. The word lists for all subject terms are placed sequentially in SI, and each subject term fills its own block of SI. Thus, the word "of" may occur more than once in the entire SI list, but only once in each of several subject terms. Each subject term is also marked with its weight number.

Two definitions help in understanding the counting procedures used in analyzing documents. First, an appearance of a word is the occurrence of that word in either the title or abstract. A word may have only one title appearance and one abstract appearance per document. Note that title appearances are counted separately from abstract appearances.

Given that a word appears in a document title or abstract, then the number of times that the word occurs in SI of the same document is defined as the usage of the word. A word may have a usage count greater than one since the word may be used in several subject terms. The usage count of a word having title appearances is kept separately from the usage count of the same word also having an abstract appearance. This distinction has been made because the significance of a word appearing in the abstract is different from the significance of the same word appearing in the title of a document.

The usage for each word is also broken down into usage by the weight of the subject terms in which it occurs. Thus, a word may have a total usage of three, with a usage of two in weight-2 terms and a usage of one in weight-4 terms. This distinction of usage by weight is made to determine if the weight usage of a word provides any clues for selecting words for a pre-index. Results indicate that weight usage is not particularly significant. Only the aggregate usage is meaningful.

At first glance it may appear somewhat arbitrary to permit a word to have only one appearance in a title or abstract. However, allowing multiple appearances would create a difficult problem in counting the usage for a particular word. With multiple appearances, it would be impossible to determine just which appearance is responsible for the usage of that word in the subject index. Moreover, the important

statistic is not so much how many tokens of a word appear in the title or abstract, but rather, given that the word appears at all, how likely is that word to be used in the subject index.

Building a Dictionary

In order to provide a future basis for judging the significance of words for pre-indexing, the data that are collected for appearances and usage of the words of the title and abstract must be stored in a dictionary. As each new document record is analyzed, the words from the title and abstract of the document, along with the appearance and usage data for those words, are added to the dictionary. Thus, the dictionary will be a list of all words that have appeared in a title or abstract to date, along with cumulative data about appearances and usage. In the data portion of the dictionary, title data and abstract data are separated in the same manner that the data were collected, although the dictionary includes each word only once.

As stated previously, 80 document records were analyzed automatically by computer in the manner just described. Then the data contained in the dictionary were inspected in order to learn about the statistical basis for the usage of words in the subject index. Note that at this stage, the data that has been collected merely represent the behavior of the human indexers.

One item of interest is the number of words that the dictionary contains, since the number of words in the dictionary may influence the usefulness of the dictionary for pre-indexing. Figure 7 shows the number of words in the dictionary as a function of the number of document records analyzed. The size of the dictionary is growing less rapidly in the region of 80 documents than it was in the first few documents of the collection. New words were added to the dictionary at the rate of approximately 35 per document during the analysis of the first 20 documents. After 80 documents, the dictionary grew at the rate of 24 new words per document. Such a growth curve indicates that the dictionary includes many common words after only a small sample of documents has been analyzed, but that new words will be encountered frequently in new documents. Later, in the design of a pre-indexing scheme for new

documents, it will be necessary to provide for those words that are not included in the dictionary, but appear in the title or abstract of a new document.

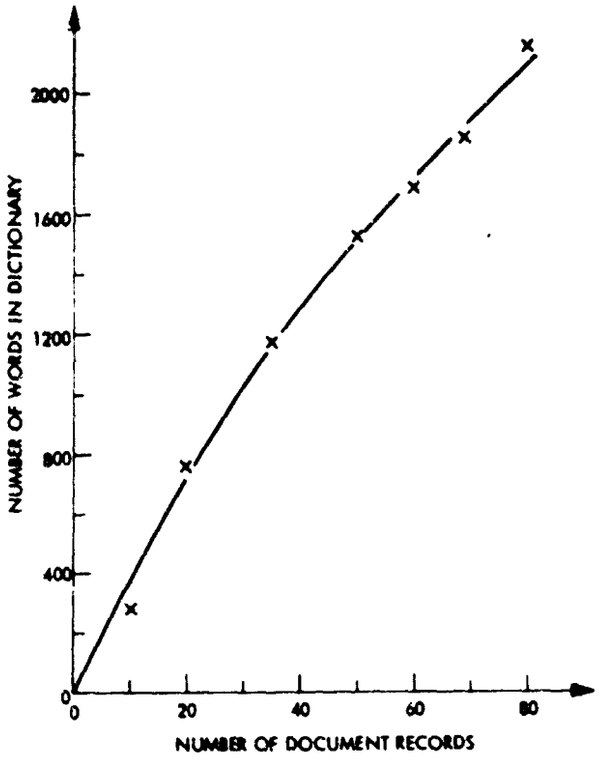


Fig. 7 Size of Dictionary as a Function of the Number of Documents Analyzed

The dictionary does not include all words that have been used in subject index terms. The dictionary is intended to provide information on the selection of words from the title and abstract. Figure 8 shows

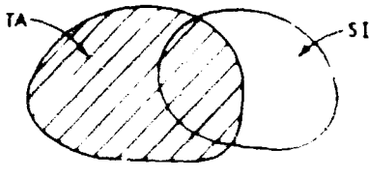


Fig. 8 Set of Words Included in the Dictionary Data Base (Shaded Area)

again the relationship of the set of title and abstract words to the set of subject index words. All the words in the shaded portion are filed in the dictionary. They include all title and abstract words. The words

in the unshaded portion are not included. Although the unshaded set of words may well be good words for inclusion in a subject index for future documents, it is not the purpose of the dictionary merely to list such words. Since the words represented by the unshaded portion of Fig. 8 are not directly connected with the words of the title and abstract, the words of the unshaded portion are omitted.

Possible Pre-Indexing Criteria

One possible clue for the selection of a word for a pre-index is the frequency of appearance of the word in document records already analyzed. The frequency of appearance of a word refers to the percentage of the document records analyzed in which the word has made an appearance. Thus if a word has appeared in 40 of 80 abstracts, then the word has a 50 percent frequency of appearance in abstracts. The same word may have a frequency of appearance of only ten percent in titles. Figure 9 shows the number of words from the dictionary in

Percentile of Frequency	Number of words in percentile	
	Title	Abstract
100	0	2
90 - 99	0	3
80 - 89	0	2
70 - 79	0	1
60 - 69	1	3
50 - 59	0	3
40 - 49	0	3
30 - 39	3	6
20 - 29	0	16
10 - 19	6	71
0 - 10	410	2009

Fig. 9 Frequency of Appearance for Words in the Dictionary Based on 80 Document Records

various percentiles of frequency for both title appearances and abstract appearances. High-frequency words tend to be function words -- mostly prepositions and articles. Such words convey little about the content of a document. Titles tend not to include high-frequency words.

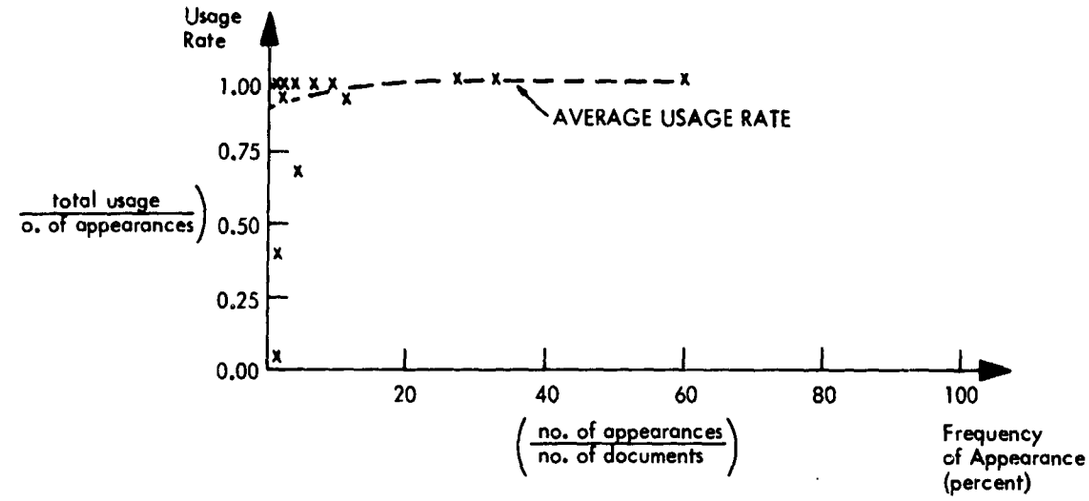
Another possible criterion for pre-indexing is the usage rate of words in the dictionary. Given that a word has appeared in a title (or abstract), then the usage rate of a word is the ratio of the number of times it is used in subject indexes to the number of appearances in titles (or abstracts). That is,

usage rate = (cumulative usage)/(no. of appearances).

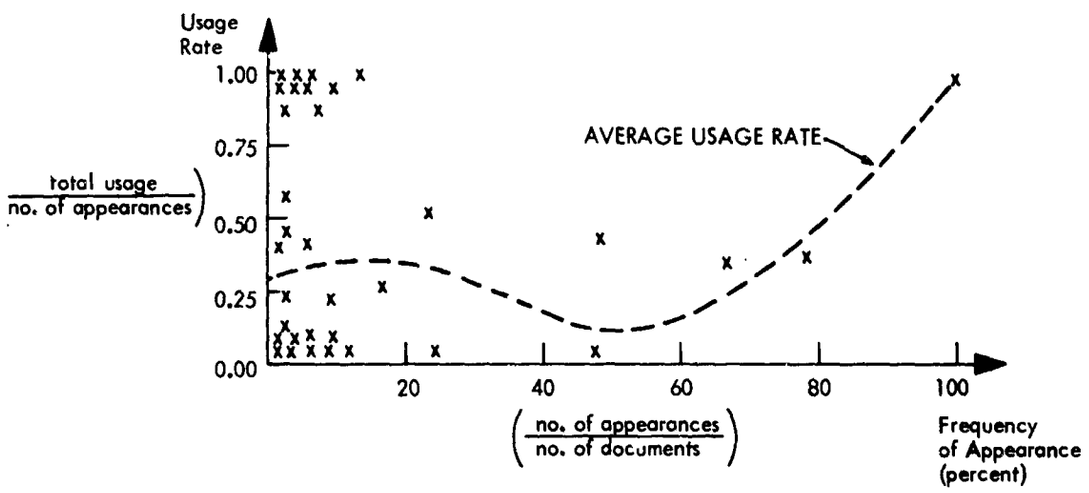
The usage rate, then, is just the average usage of the word per appearance. A word has two usage rates. One usage rate is based on its appearances in titles; the other, on its appearances in abstracts. A high usage rate for a word means that whenever the word appears in a title (or abstract) it has a high probability of being used in the subject index.

Thus, there are two main types of information available for words in the dictionary -- usage rate and frequency. Figure 10 shows the usage rate of the words in the dictionary as a function of frequency of appearance. Since the usage of a word can be higher than the number of appearances, the usage rate has been truncated at a maximum value of 1. The dashed lines in Fig. 10 are the average usage rates as a function of frequency. Each word in the dictionary is actually represented by a point somewhere on the two-dimensional graphs. (The points that are shown in the figure are merely dummy points for purposes of illustration.)

The data represented in Fig. 10 are an aggregate of the usage data for the different weight usage counts. Although the usage-count data that is stored in the dictionary is separated by weight usage, the usage rates shown in Fig. 10 are determined from the total usage counts for all weights. A graph showing usage rate by the individual weights yields similar results. With title words, however, most of the usage occurs in weight-1 subject terms. Title words have an extremely high probability of being used in subject terms -- over 90 percent. The only peculiarity of abstract word use is that high-frequency words are seldom used in terms of weight (0) or weight (4), indicating that terms of such



(a) Title Words



(b) Abstract Words

Fig. 10 Average Word Usage Rate as a Function Frequency of Appearance

weights tend to be quite specific. High-frequency "function" words, such as prepositions, are used in phrases of other weights in order to make such terms readable noun phrases, for example, "positron annihilation in potassium".

There is a useful characteristic of word usage that does not show in Fig. 10, but is clearly demonstrated in Appendix A. Very few low-frequency words have usage rates at intermediate levels near the average usage rate. The vast majority of low-frequency words are either used very seldom or very often.

Thus, the dictionary contains a large number of words that the indexers have used in the subject indexes of documents. However, the dictionary also contains a large number of words that the indexers have consistently failed to use in subject indexes. Such words include verbs, which never appear in subject terms. Hence, the dictionary contains information that will be useful when dictionary words are encountered in pre-indexing new documents. When a word encountered in a new document is found in the dictionary, the data on usage rates will indicate whether the human indexers have considered the word as useful for indexing. Of course, words that are not in the dictionary will be discovered in new documents. Rules for handling both known and new words will be necessary in generating a pre-index.

The next chapter describes the actual methods to be used for pre-indexing.

CHAPTER III

THE DEVELOPMENT OF PRE-INDEXING SCHEMES

All the automatic pre-indexes that have been evaluated have a very simple form. Basically, the pre-index is a list of words which is intended to represent the content of a document. All the pre-indexes generated are the result of simple word-by-word selection procedures to choose words from the titles and abstracts of documents. Only in one of three methods examined are words considered in context. A random sample of 30 documents with both titles and abstracts has been used for the testing of pre-indexing.

The Form of the Title and Abstract for Pre-Indexing

All pre-indexing methods tested in this research operate on a list of the title and abstract words in the document to be indexed. In preparing a title and abstract of a document for pre-indexing, we have made a slight change from the form of the TA list as described in the preceding chapter. For pre-indexing, a single list of title and abstract words in a document is prepared, but now the list gives all words in exact order of their occurrence, including multiple occurrences. Such a listing preserves information on context, which may be useful for pre-indexing.

Three Methods of Pre-Indexing

Three different algorithms for pre-indexing from a title and abstract word list have been developed. The three methods have some common features. The similarities in the methods lie in the handling of title words and in the handling of abstract words not found in the dictionary. The primary differences in the methods are in the way that the dictionary data are used.

For all methods, the words of the title of the document are included in the pre-index. The data contained in the dictionary indicates that including the title words is a very sound decision rule. From the data contained in the dictionary it is found that title words have at least a

90 percent chance of being used in the subject index of a document; hence title words should definitely be included in a pre-index.

All methods of pre-indexing face the problem of new words, that is, the occurrence in title or abstract of a word which is not contained in the dictionary. All new words, clearly, will be low-frequency words, since the new words have not appeared in any documents previously. The usage-rate averages, as graphed in Fig. 10 of Chapter II, show that a low-frequency word in an abstract has only about a 30 percent chance of being used in the human-generated subject index of the document. Thus, when a new word appears in the abstract of a document, there is no valid reason to include it in the pre-index. Nevertheless, if all new words are excluded, then many of the words that belong in the pre-index will be omitted. As noted previously, even when a dictionary comprising the words from 80 documents is used, over 20 new words are encountered in the title and abstract of a new document. Therefore, it has been decided that in all three pre-indexing methods a new word will be included <u>if it appears at least twice</u> in the title and abstract. A manual inspection of several document records indicates that if the word is used at least twice, then there is a good chance that the word represents something important to the content of the document. New words which occur only once in the title and abstract appear much more likely to be only filler and not central to the content of the document. Actual attempts at pre-indexing support the notion that only a single occurrence of a new word is insufficient for the word to be included in the pre-index.

The three methods differ mostly in the handling of words that are contained in the dictionary. The primary information available about words in the dictionary is the frequency of appearance of the word and the usage rate of the word. For example, the dictionary contains the information that the word "the" has appeared in 80 abstracts and has a cumulative usage count from abstracts of 27 in weight-1 terms, 13 in weight-2 terms, 46 in weight-3 terms, 22 in weight-4 terms, but only 1 in weight-0 terms. The three methods use this information in different ways. However, all usage data is aggregated over all subject-term weights. No use is made of the breakdown by individual weights. Thus only the aggregated usage

count of 109 is retrieved from the dictionary. From the number of appearances in abstracts and the total usage from abstracts, it is then known that the word "the" has a frequency of 100 percent in abstracts and a usage rate greater than 1.0.

Method I - Usage Rate Only

The first method considers only the usage rate for a word which is found in the dictionary. Words with a high usage rate are selected for the pre-index. Words with a low usage rate are excluded from the pre-index. Figure 11 illustrates method I. Words with usage rates falling in the shaded area are included in the pre-index for the document. The exact level of usage rate, R, that is required for a word to be included in the pre-index is tested at three different threshold levels to determine what effect the usage rate threshold has on the completeness and relevance of a pre-index. The three threshold levels of the usage rate, R, that have been tested are 0.25, 0.50, and 0.75.

Method II - Usage Rate and Frequency Thresholds

In the second method, not only the usage rate of a word is considered, but also the frequency of appearance of the word is considered. Since very high-frequency words tend to carry little information, such words may possibly be excluded from the pre-index. Therefore, in method II the pre-index will include only low-frequency, information-bearing words. Figure 12 illustrates the selection criteria for method II. Words whose frequency and usage rate fall in the shaded area are included in the pre-index. All other dictionary words are excluded. In tests of this method, the usage-rate threshold, R, has been held constant at 0.5 so that the effect of varying the frequency threshold, F, may be observed. The frequency threshold has been tested at 25 percent, 50 percent, and 75 percent.

Method III - Context Decisions

The third method of pre-indexing is somewhat more involved. In the other two methods, a word is either selected or not selected for the pre-index purely on the basis of the dictionary data for that

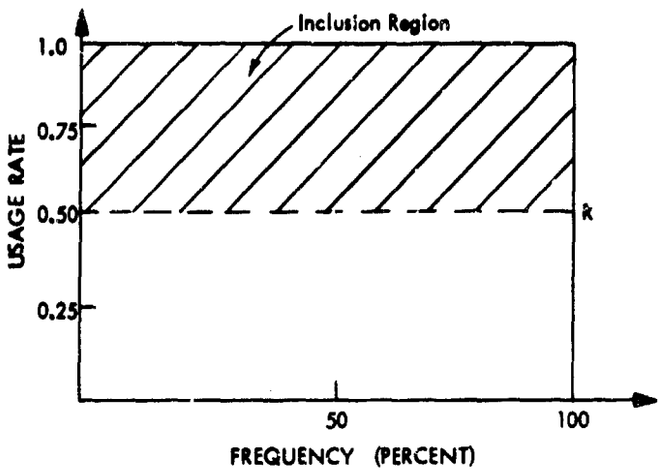


Fig. 11 Word Inclusion Region for Method I

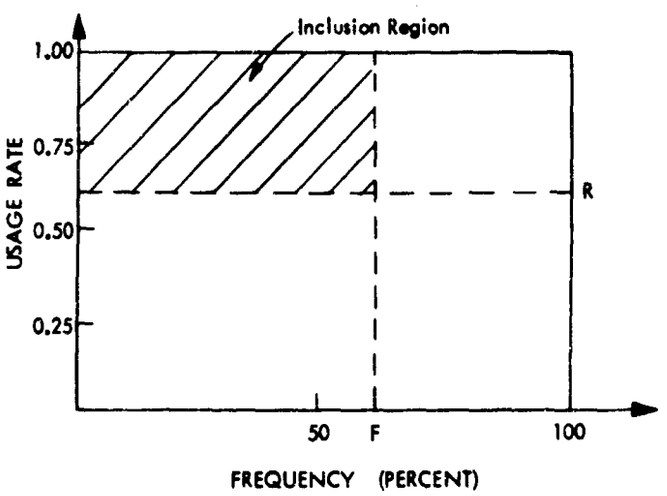


Fig. 12 Word Inclusion for Method II

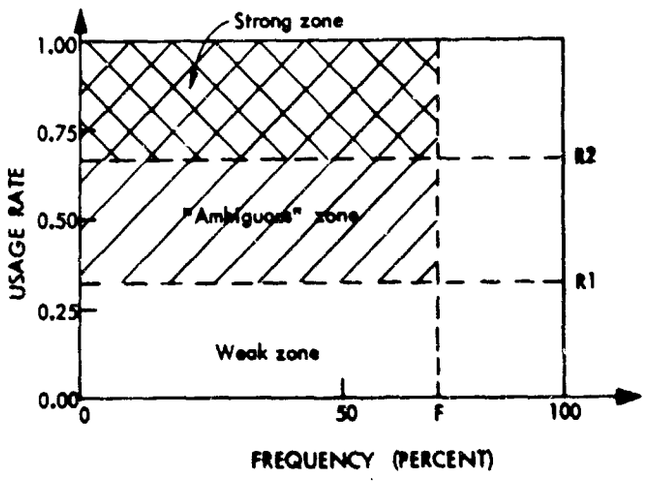


Fig. 13 Word Zones for Method III

word only. The third method of pre-indexing recognizes an "ambiguous" class of words, for which the pre-indexing method considers neighboring words before making a decision to include or exclude the ambiguous word. Figure 13 shows two thresholds, R1 and R2 on usage rate and a threshold frequency F, being used in the third method of pre-indexing. Only words with a frequency below the frequency threshold F are considered as candidates for the pre-index. Words whose usage rates fall in the zone between the two usage rate thresholds R1 and R2 are considered to be in the ambiguous zone. Weak words, which have usage rates lower than R1, are immediately excluded from the pre-index. Strong words, whose frequency and usage rates fall in the cross-hatched region above R2 in Fig. 13, are immediately included in the pre-index. Words with usage rates between R1 and R2 (the shaded region of Fig. 13) are marked for later decision. If such an ambiguous word neighbors a strong word on either side, then the ambiguous word is included in the pre-index. However, if the ambiguous word is surrounded only by weak words then the ambiguous word is excluded from the pre-index. This method was devised purely as an experiment to aid in the handling of words whose usage rate is at an intermediate level and for new words whose usage rate is unknown. For such words there is no sound basis for a selection decision, so content is considered in this method. In essence, this method recognizes simple phrases that are delimited by weak words such as articles, prepositions, and verbs. For example, from the phrase "...the dynamic pulse hysteresis in...", method III will select the words "dynamic pulse hysteresis."

In method III as in the other two methods, if a new word (a word not contained in the dictionary) occurs at least twice in the title and abstract of a document, then that word is included in the index. If such a new word occurs only once, it is considered to be an ambiguous word and is included only if it has a strong neighbor. This decision rule was adopted because a manual inspection of many abstracts indicated that although many meaningful words may appear only once in the title and abstract of a document, such words usually seem to occur next to other meaningful words. Through consideration of single-occurrence new words as ambiguous rather than weak, fewer

meaningful words are omitted from a pre-index. Later pre-indexing trials showed that the primary difference between method II and method III lies in the treatment of new words with single occurrences. Recall that such words are excluded from a method II pre-index.

In tests of method III the frequency threshold was held constant at 75 percent so that the effect of varying the ambiguous usage zone could be observed. For testing, the ambiguous zone was set at three different ranges--0.2 to 0.5, 0.3 to 0.5, and 0.3 to 0.6.

Testing the Pre-Indexing Methods

Thirty documents were automatically pre-indexed in order to test the above methods. To facilitate experimentation, the automatic pre-indexing system was developed to pre-index each document by all three methods with only one pass on a document. To provide further experimental efficiency, the pre-indexing system also pre-indexed under three different parameter sets for each method. In addition, the pre-indexing system also compared the different trial pre-indexes of each document to the human-generated subject index in order to yield immediate values of completeness and relevance for the pre-indexes. Completeness and relevance are the standards of quality as defined in the previous chapter. The next chapter describes the results of the tests of pre-indexing.

CHAPTER IV

RESULTS OF PRE-INDEXING TRIALS

To test the pre-indexing methods that were described in the preceding chapter, 30 documents having titles and abstracts were used. For each of the documents, nine different pre-indexes were generated. The nine pre-indexes are the result of using the three methods, each with three parameter sets. The testing demonstrated that the methods cause significant differences in pre-indexing results, although changing parameters within a method has little effect on the quality of a pre-index as measured by the standards of completeness and relevance.

A simple way to view the results for a pre-indexing method is to plot the completeness and relevance for each document pre-index as a point on a graph. The closer that a document pre-index is to 100 percent in both completeness and relevance, the better the pre-index, at least in a primitive way. The results of pre-indexing trials indicate that there is cause to question the measure of completeness as it has been defined previously. However, for the time being, Figs. 14, 15, and 16 show the plots of completeness and relevance for pre-indexes by methods I, II, and III respectively. Although three different parameter sets were tested for each method, a representative parameter set has been selected to illustrate the results of each method. Figure 17 summarizes the results for all three parameter sets for each of the three methods. In Fig. 17 only the average results of completeness and relevance are plotted. Appendix B gives the results of all pre-indexing trials in tabular form.

Subjective Evaluation of the Informational Content of Pre-Indexes

One of the first questions that comes to mind is, "Does an automatic pre-index appear to describe the subject content of the document?" An inspection of pre-indexes for several documents indicates that, indeed, the pre-indexes contain reasonably accurate subject matter from the title and abstract of documents. For example, from

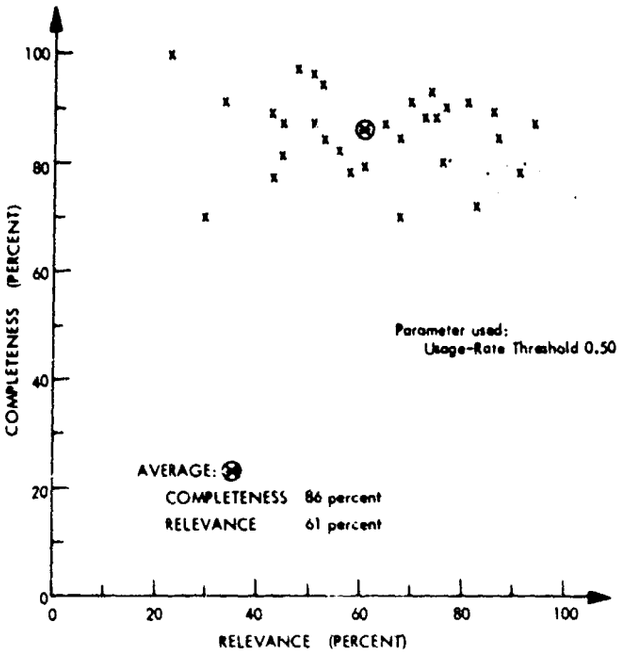


Fig. 14 Results of Method I for 30 Documents

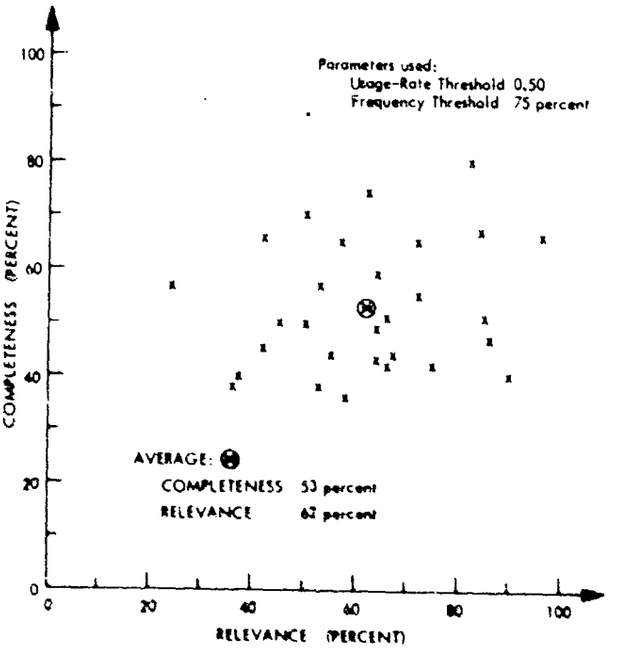


Fig. 15 Results of Method II for 30 Documents

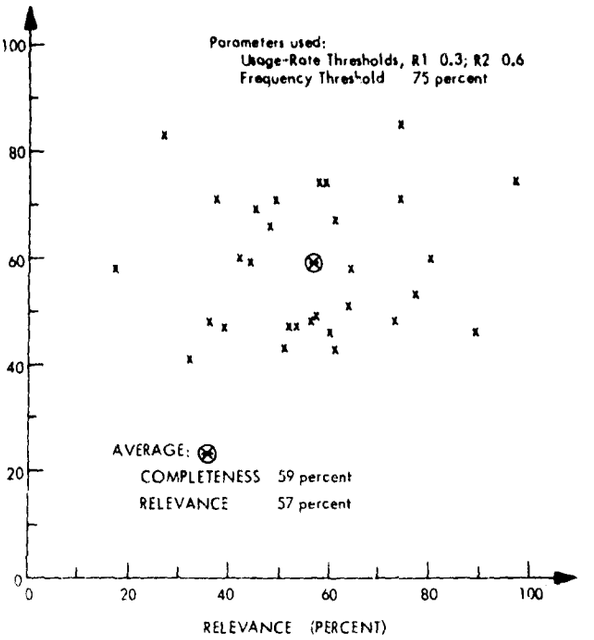


Fig. 16 Results of Method III for 30 Documents

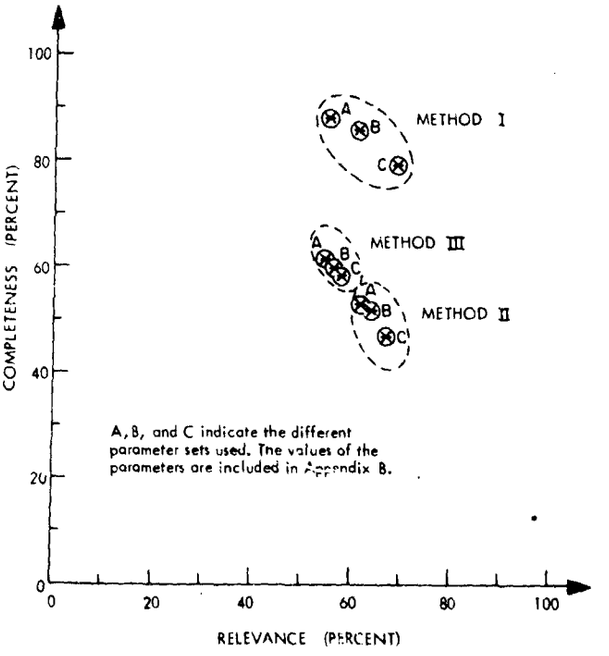


Fig. 17 Comparison of Average Results for all Methods on 30 Documents

the title and abstract of an article about "Dynamic Pulse Hysteresis in Magnetic Devices", the following is a partial list of terms in a pre-index generated by method III.

instantaneous hysteresis
magnetic
risetime
an applied pulse field
dynamic pulse hysteresis
magnetic devices
quasi-static hysteresis
magnetic damping phenomena by
extended
with

The pre-indexing method overlooked some words that are pertinent, such as "ferrite core", but in general, the pre-index contains a significant number of the important words from the title and abstract. Only a few words of the pre-index words are not pertinent to the topic, such as "an", "by", "with", and "extended". In general the pre-indexing methods are very good at eliminating verbs from pre-indexes. The word "extended" is included because it has a high usage rate from prior usage by the human indexers, who probably used it as an adjective rather than a verb. Overall, however, the pre-indexing system appears to do an effective job.

The Role of Function Words

The results from method I and method II, as illustrated in Figs. 14 and 15, apparently show that method I is far superior to method II. Both methods yield very nearly the same measured relevance; however, method I gives a completeness measure a full 33 percent higher than method II. Yet the only difference in the two methods is that method II eliminates high-frequency words from a pre-index, such as "in", "to", "for", "the", and "of". Both methods include all title words in the pre-index. Both methods have the same usage rate threshold for words found in the dictionary. Moreover, both methods treat new words in exactly the same way. But method II excludes numerous function words by excluding high-frequency words. Such function words have little informational value. In fact the two most frequent words that still carry information are "magnetic" and

'field", with frequencies just under 30 percent. Words of such low frequency have not been eliminated from pre-indexes. However, human indexers include high-frequency words in subject index terms in order to make noun phrases. (In INTREX each word of a noun phrase is being placed in an inverted file of subject words and its position within the phrase is being recorded. The purpose of this procedure is to allow users of the INTREX system to state their requests in terms of noun phrases, if they wish. INTREX plans to test and evaluate the merit of this kind of capability.) When method II eliminates such high-frequency function words from a pre-index, the completeness of that pre-index is reduced. Nevertheless, the informational value of that pre-index is unchanged, since the excluded function words carry no information. The comparison of method I with method II shows that fully 30 percent of all word occurrences that are eventually found in the human-generated subject index are merely informationless function words. Thus, the comparison of the two methods shows that the measures of completeness and relevance unfortunately do not fully indicate the quality of a pre-index. However, if the problem of high-frequency words is kept in mind, then the measures of completeness and relevance still give at least a feeling for the relative results of pre-indexing trials.

The Effect of Varying the Usage-Rate Threshold

Within both methods I and III the usage-rate thresholds are varied to determine if the exact setting of the usage rate threshold affects the pre-index significantly. In method III, where high-frequency words are excluded from the pre-index, the settings of the usage-rate thresholds have very little effect on the completeness and relevance of the pre-indexes. This is to be expected, since only five percent of all low-frequency words found in the dictionary have usage rates that fall in the intermediate range from 0.20 to 0.60, the range in which the thresholds were varied. Thus, the particular setting of the usage-rate threshold has negligible effect on the pre-indexes generated by method III. However, in method I the setting of the simple usage-rate threshold seems to have a much greater effect on the pre-indexes generated by this method. This is not surprising

when it is remembered that method I includes high-frequency words in the pre-index; their inclusion depends on the usage rates of such high frequency words. Since the usage rates of several of the higher frequency words fall in the range of trial variation of the usage rate threshold for method I, the completeness and relevance are more sensitive to the usage-rate threshold.

The Effect of New Words on Pre-Indexing

Examination of Fig. 17 shows a noticeable difference in the pre-indexing results between method II and method III. However, from the preceding discussion, it appears that there should be little difference because of the different ways of setting the usage-rate thresholds for the two methods, since both methods eliminate high-frequency words. Also, recall that both methods include new words that appear at least twice. The difference in the results obtained from the two methods arises from the different way of handling new words that occur only once in the abstract of a document. In method II, such words are considered as ambiguous words and may be included in the pre-index under the proper circumstances. By including such single-occurrence new words, method III has a higher completeness than method II, since the inclusion of any word in a pre-index can only help the measured completeness of that pre-index. On the other hand, some of the new words that method III selects for inclusion in the pre-index are not used in the human-generated subject index; hence the relevance of method II pre-indexes is lowered slightly.

In the early experimental stages of this research, all methods of pre-indexing included all new words regardless of the number of occurrences of that word in the title and abstract of the document. However, changing the decision rule to require at least two occurrences of a word for definite inclusion in a pre-index appears to have significantly increased the quality of the pre-indexes. Under all methods of pre-indexing, the relevance of pre-indexes increased by approximately eight percent, while the completeness suffered by only two to three percent when the decision rule was changed to require at least two occurrences.

The Completeness-Relevance Trade-off

Further examination of Fig. 17 clearly shows that within a pre-indexing method, relevance must be sacrificed in order to gain completeness. Moreover, comparing method II and method III, which differ most significantly in policy toward new words, one also finds that the same trade-off is made between methods. Such is to be expected. Adding any word to a pre-index can never reduce completeness. In fact if every word of the title and abstract of a document is included in the pre-index, then the completeness for that pre-index will be 100 percent. But the relevance of that pre-index will be approximately 35 percent, since only that percentage of words in the title and abstract of a document are used in the average human-generated subject index. (The analysis of document records in Phase I showed that approximately 35 percent of the words of the title and abstract eventually appear in the human-generated subject index.) When words are eliminated from a pre-index in order to increase relevance, there is the statistical probability that some of the eliminated words are actually necessary to maintain completeness. Hence, completeness must be sacrificed if relevance is to be raised. Moreover, as is found by comparing method I with method II, a full 30 percent of completeness can be sacrificed without damaging the informational content of a pre-index merely by removing the high frequency words from the pre-index.

The Effect of Dictionary Size

Even with a dictionary containing all the words from the titles and abstracts of 80 documents, many new words are encountered in the pre-indexing of new documents. An examination of the curve of dictionary growth in Fig. 7 indicates that even if the dictionary were obtained from a much larger document base, new words in titles and abstracts would continue to have an important role in pre-indexes. However, the dictionary does contain a good collection of 40 very common words, which appear in over 30 percent of all document abstracts. Also, an examination of several pre-indexes reveals that the dictionary contains most of the verbs which are encountered in typical abstracts. The three methods of pre-indexing are very good

at recognizing and excluding verbs from pre-indexes, for example, forms of "to be", "to report", "to have", and "to discuss". In fact a large portion of the selection decisions that the pre-indexing methods must make about words found in the dictionary result in the exclusion of those words, rather than the inclusion of the words. In addition, most decisions must be made about common words, since such words appear most frequently. Therefore, the dictionary used in this research is of sufficient size to provide a reasonable test of the postulated pre-indexing methods. Only a very much larger dictionary would alter the proportion of new words encountered, and it is doubtful that the results would be substantially different.

CHAPTER V

CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

The primary conclusion to be drawn from this research is that automatic pre-indexing by machine is feasible with the application of simple techniques. Furthermore, the implementation of the pre-indexing methods developed here can be accomplished with computational efficiency if the pre-indexing system is designed as an operational system rather than as an experimental system.

The first step in the design of an operational pre-indexing system is to select a suitable pre-indexing method. It is the opinion of this author that method I is not so useful for pre-indexing as are the other two methods. Recall that method I allows the inclusion of high-frequency function words in the pre-index for a document, whereas the other two methods exclude them. In a pre-index which is a list of descriptive words, function words have no place.

The remaining pre-indexing methods, II and III, present a trade-off between completeness and relevance. Method III, which includes a greater percentage of the new words that it encounters than does method II, tends to be slightly more complete, but somewhat less relevant.

However, it is necessary before one judges the relative merits of methods I and II to look more closely at the figures for completeness for both of these methods. The graph of Fig. 17 shows method II with an average completeness of 53 percent; method III, 59 percent. From the prior examination of method I, recall that approximately 30 percent of the completeness measured for a pre-index is lost if high-frequency words are removed. That particular 30 percent of completeness can be safely ignored in considering informational content, since the high-frequency words removed in methods II and III are all merely function words. Thus in terms of residual informational content, the true average completeness of methods II and III probably lies in the range of 80 percent to 90 percent, after making a correction for the extraneous 30 percent. Thus, both methods II and III appear to

contain a substantial portion of the informational content of the title and abstract of a document.

Nevertheless, method III, which includes more of the new words, would probably be the best pre-index for an operational system because of its greater completeness. The price that must be paid for the greater completeness is lessened relevance, but the larger pre-index generated by method III should not create an undue burden.

Another distinction between method II and method III is that method III recognizes and provides special processing for words having intermediate usage rates. Because a small dictionary was used in the pre-indexing tests, this difference in methods caused only very small differences in results from method II; with a greatly expanded dictionary, method III would probably become even more useful.

Streamlining the Pre-Indexing System

Much can be done to streamline the pre-indexing system for either operational use or further experimentation. Most of the possible improvements should be made in the organization of the dictionary files. Presently, the files contain an excessive amount of data that were collected in the automatic analysis of documents. At the time the data were collected, there was no way to determine just what data would be useful. From an examination of the accumulated data, however, it appears that the breakdown of usage information by subject-term weights can be discarded, and only an aggregated form of usage data retained. Such a condensation would reduce the size of the files by 35 percent. The use of sorting techniques in organizing the dictionary would improve the efficiency of the pre-indexing system. Proper organization would allow the use of faster search techniques in dictionary lookups. Such organization of the dictionary and implementation of improved search would be necessary for an operational system, or even for extensive additional research. In a pioneer experimental system with a relatively small dictionary, such refinements are not justified.

Additional streamlining can be performed on some of the programming. Several operations that at first appeared necessary for a preliminary experimental system are not clearly superfluous.

Applicability of Pre-Indexing to other Subject Areas

All of the documents used for this pre-indexing research fall in the area of materials science and engineering. Thus, the dictionary contains the special vocabulary used by scientists and engineers of that field. A question arises as to whether the dictionary and pre-indexing system developed in this research are useful in other subject areas. Unfortunately, there is no document collection from another field available for direct experimentation.

However, the nature of the pre-indexing system would probably allow it to be effective in another technical field. The pre-indexing system does not work solely by choosing recognizable words from a title and abstract; the system also eliminates many common words. In fact, most word decisions made by the system are to eliminate a word rather than to include it. Many of the words very common to materials science are function words and therefore are also very common in other fields. The pre-indexing system is good at eliminating such words, since those words are the verbs and prepositions. Hence, the dictionary would still be at least partially effective in some other field.

Suggestions for Further Research

One item of obvious interest is the effect of dictionary size on the results of pre-indexing trials. The dictionary used in these tests contained approximately 2200 words. A much larger dictionary may affect the results in two ways. First, the number of new words encountered in pre-indexing may be reduced. Second, the distribution of the usage rates of the known words may change, so that the setting of the usage-rate thresholds in the different methods may become more important. For example, in a dictionary built from 80 documents, many of the words contained in the dictionary have appeared in documents only once. Hence, the word can have a usage rate of only either 1.0 or 0.0. But if the dictionary were to be based on the

analysis of 400 documents then perhaps several of the low-frequency words would have a sufficient number of appearances, so that their usage rates could fall around 0.5. An analysis of the data contained in the dictionary as it grew to its present size indicates that any significant change in the distributions of the usage-rate data is far in the future, if in the future at all. A comparison of a dictionary based on 35 documents with a later dictionary based on 80 documents showed only a very minor change in the distribution of usage rates. In both cases, the usage rates fell primarily at the two extremes.

The effect of reducing the number of new words is also hard to predict. Again it is very probable that a very much larger dictionary is necessary to substantially affect the number of new words encountered in the title and abstract of a document. Figure 7 shows that the number of new words per new document is dropping slowly after 80 documents. In fact, a base of 400 documents might be necessary to decrease the number of new words per document from over 20 at present to about 10.

A shortage of computer time prevented tests with an expanded data base. However, first a streamlining of the pre-indexing system to a more operational form, followed by testing with a larger dictionary, would no doubt be worthwhile as a prelude to making such a pre-indexing system operational.

The Possibility of Using Word Stemming

The value of stemming in a pre-indexing system was not tested during these experiments. However, stemming may have two very useful effects on a pre-indexing system. One immediate effect is to reduce the size of the dictionary that is stored as the data base. Many different forms of a root with various endings are presently stored separately in the dictionary. However, by stemming, all words that have the same root can be lumped together in one dictionary entry. For example, the entries for "magnetic" and "magnetism" can be merged. Such a consolidation of data significantly reduces storage requirements.

Moreover, the stemming of words also decreases the number of new words encountered in a new document. With stemming, a new

ending on a known root would be considered as merely another instance of the known root, rather than as a completely new and unknown word.

APPENDIX A

BREAKDOWN OF WORDS IN DICTIONARY BY USAGE RATE AND FREQUENCY

(Based on words from 80 documents)

Usage Rate	0 to 9	10 to 19	20 to 29	30 to 39	40 to 49	50 to 59	60 to 69	70 to 79	80 to 89	90 to 99	100 to
≥1.00	368	6	-	3	-	-	1	-	-	-	-
0.90 - 0.99	-	-	-	-	-	-	-	-	-	-	-
0.80 - 0.89	-	-	-	-	-	-	-	-	-	-	-
0.70 - 0.79	1	-	-	-	-	-	-	-	-	-	-
0.60 - 0.69	1	-	-	-	-	-	-	-	-	-	-
0.50 - 0.59	3	-	-	-	-	-	-	-	-	-	-
0.40 - 0.49	-	-	-	-	-	-	-	-	-	-	-
0.30 - 0.39	-	-	-	-	-	-	-	-	-	-	-
0.20 - 0.29	-	-	-	-	-	-	-	-	-	-	-
0.10 - 0.19	-	-	-	-	-	-	-	-	-	-	-
0.00 - 0.09	37	-	-	-	-	-	-	-	-	-	-

Frequency of Appearance (percent)

(a) Title Words

Usage Rate	0 to 9	10 to 19	20 to 29	30 to 39	40 to 49	50 to 59	60 to 69	70 to 79	80 to 89	90 to 99	100 to
≥1.00	660	20	4	-	1	-	1	-	-	1	2
0.90 - 0.99	-	-	-	-	-	-	-	-	-	-	-
0.80 - 0.89	4	2	-	-	-	-	-	-	-	-	-
0.70 - 0.79	9	3	-	-	-	-	-	-	-	-	-
0.60 - 0 69	22	4	1	1	-	-	-	-	-	-	-
0.50 - 0.59	52	3	2	-	-	-	-	-	-	2	-
0.40 - 0.49	6	4	-	-	1	-	1	1	-	-	-
0.30 - 0.39	25	5	-	-	-	-	-	-	-	-	-
0.20 - 0.29	15	10	-	-	1	-	-	-	-	-	-
0.10 - 0.19	6	1	-	-	-	-	-	-	-	-	-
0.00 - 0.09	1210	19	9	5	-	3	1	-	1	-	-

Frequency of Appearance (percent)

(b) Abstract Words

APPENDIX B

RESULTS OF PRE-INDEXING TRIALS

C = Completeness in percent
R = Relevance in percent

Method I

	Doc No.	A Usage rate threshold: 0.25		B 0.50		C 0.75	
		C	R	C	R	C	R
1	300	82	72	80	76	74	84
2	302	93	80	89	86	73	88
3	304	87	86	84	87	85	89
4	305	89	92	87	94	82	100
5	364	78	88	78	91	75	96
6	383	97	45	97	48	94	58
7	401	96	48	96	51	96	59
8	402	100	19	100	23	100	33
9	415	90	41	87	45	87	57
10	437	72	63	70	68	76	73
11	606	88	69	88	73	86	87
12	619	91	79	91	81	87	86
13	620	87	43	87	51	87	58
14	621	78	55	78	58	64	57
15	622	78	82	72	83	71	92
16	623	75	29	70	30	70	36
17	625	84	43	82	56	77	52
18	719	91	51	84	53	80	61
19	721	83	56	79	61	76	69
20	722	90	41	89	43	59	37
21	723	94	52	94	53	94	64
22	724	94	71	90	77	81	79
23	818	97	65	87	65	81	78
24	905	93	73	93	74	65	74
25	907	81	43	81	45	63	49
26	908	95	67	91	70	78	77
27	909	89	45	77	43	73	53
28	911	91	29	91	34	91	44
29	1102	92	65	84	68	80	80
30	1106	88	75	88	75	79	90
Average		88	55	86	61	79	69

Method II

		A		B		C	
	Frequency threshold:	25%		50%		75%	
	Usage rate threshold:	0.50		0.50		0.50	
	Doc No.	C	R	C	R	C	R
1	300	40	74	42	72	42	66
2	302	34	74	39	74	42	75
3	304	45	89	45	86	51	85
4	305	53	95	61	96	66	96
5	364	42	87	45	85	47	86
6	383	60	68	71	71	71	61
7	401	57	61	57	58	57	53
8	402	36	20	57	27	57	24
9	415	65	72	65	54	65	72
10	437	32	63	36	59	36	58
11	606	46	76	51	73	51	66
12	619	38	69	44	70	44	67
13	620	58	64	65	51	65	57
14	621	41	64	41	63	49	64
15	622	34	96	37	93	40	90
16	623	55	58	70	58	70	50
17	625	41	43	43	42	45	42
18	719	40	47	46	49	50	50
19	721	47	79	55	80	55	72
20	722	35	40	38	38	40	37
21	723	65	69	74	66	74	62
22	724	30	58	41	62	43	64
23	818	42	62	44	60	44	55
24	905	65	87	65	79	80	82
25	907	38	48	38	39	38	36
26	908	38	62	38	56	38	53
27	909	48	63	50	50	50	45
28	911	66	58	66	45	66	42
29	1102	57	80	59	74	59	64
30	1106	63	88	67	89	67	84
	Average	47	67	52	64	53	62

Method III

		A		B		C	
	Frequency threshold:	75%		75%		75%	
	"ambiguous" zone:	0.2 to 0.5		0.3 to 0.5		0.3 to 0.6	
	Doc No.	C	R	C	R	C	R
1	300	47	57	46	58	46	60
2	302	49	70	48	71	48	73
3	304	60	79	60	80	60	80
4	305	74	97	74	97	74	97
5	364	53	77	53	77	53	77
6	383	74	54	74	55	74	58
7	401	59	41	59	43	59	44
8	402	57	17	57	17	57	17
9	415	69	43	69	45	69	45
10	437	49	57	49	57	49	57
11	606	54	60	54	60	51	64
12	619	47	58	47	69	47	53
13	620	74	50	71	49	71	49
14	621	56	58	53	57	48	56
15	622	49	85	47	86	46	89
16	623	90	43	90	44	83	27
17	625	52	36	52	36	48	36
18	719	60	40	60	41	60	42
19	721	58	62	58	64	58	64
20	722	46	31	46	33	41	32
21	723	77	60	77	59	74	59
22	724	51	55	48	63	43	61
23	818	50	50	49	50	47	52
24	905	85	72	85	74	85	74
25	907	49	35	47	36	47	39
26	908	43	47	43	48	43	51
27	909	66	45	66	47	66	48
28	911	71	32	71	33	71	37
29	1102	67	59	67	60	67	61
30	1106	71	74	71	74	71	74
	Average	60	55	60	56	59	57

APPENDIX C

PROGRAM STRUCTURE AND LISTINGS

The programs for both Phase I and Phase II of the pre-indexing system make extensive use of pointers in order to make data available to all subroutines. The programs are written in the AED language, which has convenient facilities for handling pointers.

The programs are listed and further described by phase. The Phase I programs are used to analyze document subject indexes and generate the dictionary files. Phase II actually generates pre-indexes by the three different methods.

All programs are designed to run on the Compatible Time-Sharing Computer System at M.I.T. (CTSS).

PHASE I

The main program, MAIN1, sets up all the data handling arrays. MAIN1 makes room for a list of title and abstract words, a list of subject index words, and an array for storing appearance and usage data. Pointers to the various arrays are stored in a directory array PTRS. Thus most major subroutine calls need only the argument PT, which is a pointer to the array PTRS. PTRS also contains other data which is useful to the subroutines. A list of the important entries in PTRS follows:

0 - Location of TA, the array for title and abstract words.

1 - Location of SI, the array for subject index terms.

2 - Location of D, the array for appearance and usage data.

3 - TL, the length of the title section of TA. This is entered by a subroutine.

4 - AL, the length of the abstract section of TA.

5 - A scratch location sometimes used to pass arguments in subroutine calls.

6 - The number of the current document being operated on.

8 - Another scratch location.

9 - The word length in a dictionary lookup.

10 - The file number in a dictionary lookup.

11 - Another scratch location.

12 - Location of BOOK FILE.

13 - Location of dictionary file in use.

Three major arrays are used in the analysis of a document record. These arrays are TA, SI, and D. TA contains the words of the title and abstract. SI contains the subject terms. D is filled with the usage and appearance data for the words in TA.

The TA Array

The words in the TA array are stored four characters per computer word, left justified and blank-padded. Thus a six-character word will fill two computer words and have two blank characters in the final two character positions. The words stored in TA are separated by a computer word of four blank ASCII characters.

The length of a stored word is defined as the number of computer words it fills, including the fence of blanks which separate it from the next word. Thus, a four character word such as 'film" has a length of 2--one computer word for the characters plus one computer word for the fence of blanks. The words "magnetic" and field have the same length of 3.

The length of an array of words is defined as the sum of the lengths of all individual words in the array. The length of an array does not specify the number of words in an array.

The SI Array

The first computer word of the SI array is a header which tells the number of subject terms in SI. The header is followed by the subject terms. Each subject term also has a one computer word header which contains the length of the subject term in the decrement and the weight of the subject term in the address.

The D Array

The D array is used for recording the usage and appearance data for the words in the TA array. Three computer words in D are required for each language word in TA. For the nth word in TA the

contents of D are as follows (the position of data within a word is indicated by the octal mask):

D(3N)	- Weight 0 usage from title, 7C29
	- Weight 1 usage from title, 1777C
	- Weight 2 usage from title, 777C20
	- Weight 4 usage from title, 1777C10
D(3N+1)	- Weight 0 usage from abstract, 7C29
	- Weight 1 usage from abstract 1777C
	- Weight 2 usage from abstract, 777C20
	- Weight 4 usage from abstract, 1777C10
D(3N+2)	- Weight 3 usage from title, 777C27
	- Weight 3 usage from abstract, 777C9
	- Title appearance, 777C18
	- Abstract appearance, 777C

Data are placed in the D array in this particular manner so that it is formatted for direct addition to the word data already contained in the dictionary files.

Dictionary Files

The dictionary files contain all words encountered in titles and abstracts, as well as the cumulative usage and appearance data for each word. The files are sorted only by word length. Each file is two tracks (864 computer words) long. Thus, there is a series of files for each possible word length. A file is specified by two names, such as...M03 ...N04. The first name indicates the length of words contained in the file, in this case 3. The second name indicates the number of the file in the series of files for the given word length. In this case the file is the fourth of the dictionary files having a word length of three.

For storage, each word in a file is followed directly by three computer words containing data about appearances and usage. The data words have exactly the format described for data storage array D.

The Book File

With the dictionary files as described above, a bookkeeping system is necessary to keep track of the number of files in each word length series, and to monitor the number of words stored in the last number file of each series. The Book File maintains this vital information. If a file word length is M, then the number of files in the series is stored in location 2M and the number of words contained in the last file of the series is contained in location (2M+1) of the Book File.

Operating Phase I

To operate Phase I, all Phase I programs must be loaded and started from a CTSS console. The files CATDIR FILE and CATREC FILE, which contain the document records, must be available. The program will type the number of documents analyzed to date, then request the number of a new document to be analyzed. The user responds by typing a four digit document number for the new document to be analyzed. The computer will signify when the analysis is complete and wait for another document number. The Phase I programs will not analyze a document which is missing a title, abstract, or subject index field.

Phase I automatically collects all data and builds the dictionary files.

```
      M5806      CMFL01       MAIN1       ALGOL FOR   M5806        4976       051568
BEGIN
          INTEGER ARRAY TA(1200),SI(9CC),DI9CO),A(8),BK(432),PTRS(15) $,
          INTEGER PT,EOF,EOFCT,I,BOOK,FILE,RW,N,Z,BKPT $,
          PROCEDURE CMNCOM,IGTFLC,OPEN,WRWAIT,CLOSE $,
          PROCEDURE WCOUNT,CCT,FSTATE,RDWAIT $,
          INTEGER PROCEDURE SET $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          PRESET BEGIN
            BOOK .BCD. /  BCOK/ $,
            FILE .BCD. /  FILE/ $,
            RW .BCD. /RW/ $,
          END $,
          PT=LOC PTRS $,
COMMENT INITIALIZE PTRS ARRAY WITH ALL DIRECTORY INFORMATION $,
          PTRS(C)=LOC TA(1) $,
          PTRS(1)=LOC SI(1) $,
          PTRS(2)=LOC D $,
          PTRS(12)=LOC BK $,
          IGTFLC() $,  ... INITIALIZE THE CATALOG FILES TO BE USED//
COMMENT PLACE BOOK FILE INTO ACTIVE ARRAY $,
          OPEN(RW,BOOK,FILE,0,2) $,
          FSTATE(BCOK,FILE,A TO 8) $,
          IF A(C) EQL 432 THEN ROWAIT(BOOK,FILE,C,BK TO 432,EOF,EOFCT) $,
          PRINT F1,BK(1) $,  ... PRINT NO. OF DOCS. ANALYZED TO DATE//
F1$       FORMAT(/28H NUMBER OF DOCUMENTS TO DATE,I5) $,
          PRINT C1 $,  .... REQUEST DOC NO. FOR NEXT DOC//
C1$       FORMAT (32H TYPE NUMBER OF NEW DOCUMENT ***) $,
START$    READ F2,PTRS(6) $,  ... READ FOUR DIGIT DOC NO.//
          IF PTRS(6) EQL 0 THEN GOTO FINISH $,
COMMENT IF DOC NO. IS ZERO THEN STOP $,
F2$       FORMAT(I4) $,
NXCCC$    Z=SET(PT) $,  ... SET UP ARRAYS TA AND SI//
          IF Z GR1 0 THEN BEGIN
            PRINT F,PTRS(6),Z $,
F$          FORMAT(14H DOC.I4,14H MISSING FIELD,I3) $,
COMMENT PRINT WARNING AND ABORT IF ANY FIELDS ARE MISSING $,
            GOTO START $,
          END $,
          WCOUNT(ST) $,  ... MAKE COUNTS OF APPEARANCES AND USAGE//
          CCT(PT) $,  ... STORE DATA IN DICTIONARY//
          BK(1)=BK(1)+1 $,  ... INCREMENT NO. OF DOCS ANALYZED//
          PRINT F3,PTRS(6) $,
F3$       FORMAT(2HDOC,I5) $,
          GOTO START $,
FINISH$   PRINT F1,BK(1) $,
COMMENT PRINT NO. OF DOCS ANALYZED TO DATE AND CLOSE OPEN FILES
BEFORE STOPPING $,
          WRWAIT(BOOK,FILE,1,BK TO 432,EOF,EOFCT) $,
          CLOSE(BOOK,FILE) $,
          CMNCOM() $,
END FINI
```

```
      M5806      CMFL01       SEC1        ALGOL FOR   M5806        4976       051568
BEGIN
DEFINE PROCEDURE CCT(PT) WHERE INTEGER PT TOBE
BEGIN
COMMENT THIS PROCEDURE SEARCHES THE DICTIONARY FILES FOR ALL
WORDS CONTAINED IN ARRAY TA.  THE DATA FOR THE WORDS ARE ADDED TO
THE DICTIONARY AND ANY NEW WORDS ARE APPENDED TO THE
DICTIONARY BY THE ROUTINE FSRCH. $,
          INTEGER WRD1,WRD2,M,M1,M2,OPEN,CLOSE,PTEND,PTX,PTXX,ASTER $,
          INTEGER DICT $,
          INTEGER PROCEDURE WLNGTH $,
          PROCEDURE OCTFIL,FSRCH $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          PRESET BEGIN
          OPEN .BCD. /  OPEN/ $,
          CLOSE .BCD. / CLOSE/ $,
          DICT .BCD. /  DICT/ $,
          ASTER=C40040040052C $,
          END $,
          PTEND=W(PT)+W(PT+3)+W(PT+4) $,  ... END OF TA//
START0$   W(PT+1C)=1 $,  ... SET FILE NO. TO 1//
          P=0 $,
          WRD1=C $,
          PTX=W(PT) $,  ... SET POINTER TO FIRST WORD OF TA//
START1$   PTX=PTX+M $,  ... PT TO NEXT WORD//
          WRD1=WRC1+3 $,
          IF PTX GEQ PTEND THEN GOTO RETURN $,
          M=WLNGTH(PTX) $,
          IF W(PTX) EQL ASTER THEN GOTC START1 $,
          W(PT+9)=M $,  ... SET WORD LENGTH FOR DICTIONARY FILE//
START $   CCTFIL(DICT,OPEN,PT) $,  ... OBTAIN SPECIFIED DICTIONARY FILE//
FS$       W(PT+11)=WRD1 $,  ... SET DATA LOC IN ARRAY D//
COMMENT SEARCH THE CURRENT DICTIONARY FILE FOR WORD POINTED TO BY PTX $,
          FSRCH(PTX,PT) $,
COMMENT IF THE WORD IS FOUND IN THE DICTIONARY FILE THEN IT IS REPLACED
BY AN ASTERISK AND PTX IS MOVED TO POINT TO THE NEXT WORD OF THE
SAME LENGTH M $,
          IF W(PTX) EQL ASTER THEN BEGIN
             M1=M $,
NWRD$        PTX=PTX+M1 $,
             WRD1=WRD1+3 $,
             IF PTX GEQ PTEND THEN BEGIN
               CCTFIL(DICT,CLOSE,PT) $,
               GOTO START0 $,
             END $,
             M1=WLNGTH(PTX) $,
             IF W(PTX) EQL ASTER THEN GOTO NWRD $,
             IF M1 EQL M THEN GOTO FS $,
             GOTO NWRD $,
          END $,
          PTXX=PTX $,
COMMENT LOOK FOR THE NEXT WORD HAVING LENGTH M $,
          WRD2=WRC1 $,
          M2=M $,
LOOP$     PTXX=PTXX+M2 $,
          WRD2=WRC2+3 $,
          IF PTXX GEQ PTEND THEN BEGIN
COMMENT IF THE END OF THE ARRAY IS REACHED THEN CLOSE CURRENT FILE
AND GO GET THE NEXT NUMBERED FILE IN ORDER TO CONTINUE THE SEARCH. $,
```

```
            GETFILEDICT,CLOSE,PT) $,
            GOTO START2 $,
          END $,
          P2=WLNGTH(PTXX) $,
COMMENT IF A WORD HAS BEEN REPLACED BY AN ASTERISK THEN PASS IT $,
          IF W(PT+3) EQL ASTER THEN GOTO LOOP $,
          IF P2 EQL M THEN BEGIN
COMMENT IF A WORD IS FOUND OF LENGTH P THEN SEARCH THE
ACTIVE DICTIONARY FILE $,
            W(PT+11)=WRD2 $,
            FSRCH(PTXX,PT) $,
          END $,
          GOTO LOOP $,
END $,
DEFINE INTEGER PROCEDURE SET(PT) WHERE INTEGER PT TOBE
BEGIN
          INTEGER PROCEDURE GETFLD,TRNSI,TRNFER $,
          INTEGER DN,FN,TL,AL,I $,
          INTEGER ARRAY XX(10) $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          AA=9 $,
          CN=W(PT+6) $,  ... OBTAIN THE CURRENT DOC NO.//
          FA=24 $,  ... SET FIELD NO. TO TITLE//
          W(PT+5)=W(PT) $,  ... SPECIFY WHERE TITLE TO BE PUT//
          W(PT+8)=GETFLD(DN,FN,XX) $,  ... GET TITLE//
          IF W(PT+8) LES 0 THEN GOTO FINISH $,
          W(PT+3)=TRNFER(PT) $,  ... MOVE TITLE FROM BUFFER TO TA//
          FN=71 $,
          W(PT+5)=W(PT)+W(PT+3) $,  ... SET WHERE ABSTRACT TO BE PUT//
          W(PT+8)=GETFLD(DN,FN,XX) $,
          IF W(PT+8) LES 0 THEN GOTO FINISH $,
          W(PT+4)=TRNFER(PT) $,  ... MOVE ABSTRACT FROM BUFFER TO TA//
          IF W(PT+4) EQL 0 THEN GOTO FINISH $,
FN73$     FA=73 $,
          W(PT+5)=W(PT+1) $,  ... SET WHERE SUBJECT INDEX TO BE PUT//
          W(PT+8)=GETFLD(DN,FN,XX) $,
          IF W(PT+8) LES 0 THEN GOTO FINISH $,
          I=TRNSI(PT) $,  ... MOVE SUBJECT INDEX FROM BUFFER TO SI//
          IF I GR) 880 THEN GOTO FINISH $,
          FN=0 $,  ... IF NO ERROR CODES ENCOUNTERED RETURN A ZERO//
FINISH$   SET=FA $,  ... RETURN NO. OF ANY FIELD IN ERROR//
END $,
END FINI
```

```
**********************************************************************
     M5806      CMFL01         TRAN       ALGOL FOR   M5806        4976       051568
BEGIN
DEFINE INTEGER PROCEDURE WNLP(PT) WHERE INTEGER PT TOBE
COMMENT THIS PROCEDURE DETERMINES THE NUMBER OF WORDS
IN THE TITLE SECTION OF ARRAY TA $,
BEGIN
          INTEGER PTT,M,WROT $,
          INTEGER PROCEDURE WLNGTH $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          WROT=0 $,
          PTT=W(PT) $,  ... GET STARTING LOC OF TA//
COMMENT IF TITLE SECTION IS EXCEEDED THEN RETURN $,
LOOP$     IF PTT GEQ W(PT)+W(PT+3) THEN GOTO FINISH $,
          M=WLNGTH(PTT) $,
          PTT=PTT+M $,
          WROT WROT+3 $,
          GOTO LOOP $,
FINISH$   WNUM=WROT $,  ... RETURN NUMBER OF WORDS=3//
END $,
DEFINE INTEGER PROCEDURE WLNGTH(PTT) WHERE INTEGER PTT TOBE
COMMENT TO DETERMINE THE LENGTH OF A WORD POINTED AT BY
PTT. THE LENGTH OF THE WORD EXTENDS TO THE NEXT WORD
AND INCLUDES ANY INTERVENING BLANK WORDS $,
BEGIN
          INTEGER M $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          M=0 $,
LOOP1$    IF W(PTT+M) EQL 04C04C04004CC THEN GOTO LOOP2 $,
          M=M+1 $,
          GOTO LOOP1 $,
LOOP2$    M=M+1 $,
          IF W(PTT+M) EQL 04004C04004CC THEN GOTO LOOP2 $,
          WLNGTH=M $,
END $,
DEFINE PROCEDURE RPLACE(PTX) WHERE INTEGER PTX TOBE
COMMENT TO REPLACE A WORD WITH A SPECIAL * MARKER FOLLOWED
BY BLANKS UNTIL THE NEXT WORD OF THE ARRAY.  PTX MUST POINT
TO THE START OF THE WORD AND CONTAIN ITS LENGTH IN THE
DECREMENT. $,
BEGIN
          INTEGER M,I $,
          INTEGER COMPONENT W $,
          W $=$ 0 $,
          M=PTX .AS. 18 $,
          W(PTX)=C40040040052C $,
          I=1 $,
START$    W(PTX+I)=04004004C040C $,
          I=I+1 $,
          IF I LES M THEN GOTO START $,
END $,
DEFINE INTEGER PROCEDURE TRNFER(PT) WHERE INTEGER PT TOBE
COMMENT THIS PROCEDURE MOVES A FIELD FROM A BUFFER TO AN
ARRAY IN THE STANDARD FORMAT FOR THE PRE-INDEXING SYSTEM.
PTRS(1) MUST CONTAIN A STANDARD INDEX POINTER TO THE BUFFER.
PTRS(2) MUST CONTAIN A POINTER TO THE ARRAY. $,
BEGIN
          INTEGER I,K,M,L,XL,CH,PTR,PTX,PTXX $,
          INTEGER PTTEMP,BLANKW,BLANKS,CHAR,NUM $,
```

```
          INTEGER ARRAY TEMP(9),BRLIST(30) $,
          INTEGER PROCEDURE GET,WCHECK $,
          PROCEDURE INC $,
          INTEGER COMPONENT W $,
          SWITCH S=SW1,SW2 $,
          W $=$ C $,
          BLANKW=C4004004004CC $,
          PTTEMP=LOC TEMP(1) $,
          XL=0 $,
          PTR=W(PT+8) $,  ... GET PTR TO BUFFER//
          PTSI=W(PT+5) $,  ... GET PTR TO ARRAY//
          NUM=PTR .RS. 18 $,
          NUM=NUP .A. 77777C $,
START$    L=27 $,
COMMENT INITIALIZE VARIABLES $,
          P=1 $,
          M=1 $,
          FOR I=1,2,3,4,5,6,7,8 DO TEMP(I)=C $,
START.S   IF NUP LEQ 0 THEN GOTO FINISH $,  ... IS BUFFER EXHAUSTED//
          CHAR=GET(PTR) $,  ... GET NEXT CHAR//
          INC(PTR,1) $,
          NUM=NUP-1 $,
          GOTO SW1 $,
SW1$      IF CHAR GEQ 101C THEN BEGIN
COMMENT THIS SECTION CHECKS TO SEE IF THIS CHAR IS A LETTER.
IF SO THEN IT IS MADE LOWER CASE.  IF IT IS A BREAK CHAR, IT IS
DISCARDED. ASTERISKS WILL BE HANDLED SPECIALLY $,
            IF CHAR LEQ 140C THEN CHAR=CHAR+C40C $,
            IF CHAR LEQ 172C THEN GOTO BUILD $,
            GOTO FINISH $,
          END $,
          IF CHAR LEQ 051C THEN GOTO FINISH $,
          IF CHAR GEQ 053C THEN GOTO FINISH $,
          IF CHAR EQL 052C THEN BEGIN
COMMENT IF AN ASTERISK IS ENCOUNTERED, THEN ALL TYPES OF CHARACTERS
WILL BE INCLUDE IN THE WORD BEING CONSTRUCTED UNTIL THE NEXT ASTERISK
IS ENCOUNTERED. A FLAG IS SET TO MARK THESE SPECIAL WORDS. $,
            K=2 $,
            GOTO BUILD $,
          END $,
SW2$      IF CHAR EQL 052C THEN K=1 $,
COMMENT RESET SWITCH WHEN SECOND ASTERISK FOUND $,
BUILD$    CHAR=CHAR .LS. L $,
COMMENT THIS SECTION BUILDS A LEFT JUSTFIED WORD IN
TEMPORARY ARRAY $,
          TEMP(P)=TEMP(M)+CHAR $,
          IF L EQL 0 THEN BEGIN
            L=27 $,
            M=M+1 $,
            IF P GRT 8 THEN BEGIN
COMMENT IF THE WORD IS TOO LONG THEN PRINT A WARNING $,
              PRINT F1,W(PT+6) $,
F1$           FORMAT(17H LONG WORD IN COC,I5) $,
              GOTO START $,
            END $,
            GOTO START1 $,
          END $,
          L=L-9 $,
          GOTO START1 $,
FINISH$   IF L NEQ 27 THEN GOTO CONT $,
          IF M EQL 1 THEN BEGIN
            IF NUP GRT 0 THEN GOTO START ELSE GOTO DONE $,
          END $,
          GOTO BYPASS $,
CONT$     L=27-L $,
COMMENT PAC THE FINISHED WORD WITH BLANKS $,
          BLANKS=BLANKW .RS. L $,
          TEMP(M)=TEMP(M)+BLANKS $,
          M=M+1 $,
BYPASS$   TEMP(P)=BLANKW $,
          W(PT+5)=PTX+XL .LS. 18 $,  ... INITIALIZE FOR WCHECK//
          W(PT+11)=PTTEMP+M .LS. 18 $,
          CK=WCHECK(PT) $,  ... SEE IF WORD HAS ALREADY APPEARED//
          IF CK GRT 0 THEN GOTO START $,
          PTXX=PTX+XL $,
          I=0 $,
STOR$     IF I EQL M THEN BEGIN
COMMENT ADD WORD TO ARRAY $,
            XL=XL+M $,  ... INCREMENT LENGTH OF ARRAY//
            IF XL GRT 1150 THEN BEGIN
COMMENT IF ARRAY MEANS BOUND THEN ABORT AND PRINT WARNING $,
              PRINT F2,W(PT+6) $,
F2$           FORMAT(19H FIELD TOO LONG COC,I5) $,
              XL=0 $,
              GOTO DONE $,
            END $,
            IF NUP LEQ 0 THEN GOTO DONE $,
            GOTO START $,
          END $,
          W(PTXX+I)=TEMP(I+1) $,
          I=I+1 $,
          GOTO STOR $,
DONE$     TRNFER=XL $,  ... RETURN LENGTH OF FILLED ARRAY//
ENC $,
DEFINE INTEGER PROCEDURE TRNSI(PT) WHERE INTEGER PT TOBE "
COMMENT SET UP THE SI ARRAY. THE HEADER OF SI CONTAINS THE NUMBER
OF SUBJECT TERMS IN SI.  A HEADER FOR EACH SUBJECT TERM CONTAINS
THE WEIGHT OF THE TERM AND THE LENGTH OF THE TERM. A STANDARD INTREX
POINTER TO THE  BUFFER CONTAINING THE SUBJECT INDEX MUST BE
CONTAINED IN PTRS(8). $,
BEGIN
          INTEGER PTR,BKPT,XL,PTSI,PHR,PTX,LEN,WT,SIL,SIL1,LEN1,PTS $,
          INTEGER MASK,LEN2,TLEN,LENLIM $,
          INTEGER PROCEDURE WEIGHT,TRNFER $,
          PROCEDURE INC $,
          INTEGER COMPONENT W $,
          W $=$ 0 $,
          MASK=777777C $,
          XL=1 $,
          TLEN=0 $,
          PHR=0 $,
          PTR=W(PT+8) $,
          PTSI=W(PT+1) $,
          BKPT=W(PT+12) $,
          LENLIM=PTR .RS. 18 $,
          LENLIM=LENLIM .A. 77777C $,
          PTS=PTSI+1 $,
START$    WT=WEIGHT(PTR) $,  ... FIND WEIGHT AND LENGTH OF NEXT TERM//
          IF WT EQL 0 THEN GOTO FINISH $,
          LEN=WT .RS. 18 $,
          TLEN=TLEN+LEN $,
          PTX=PTR .A. MASK $,
```

```
          IF J EQL 2 THEN GOTO WCHCK1 $,
          IF CHAR NEQ CLIST(1) THEN GOTO START1 ELSE GOTO START2 $,
WCHCK1$   J=0 $,
WCHCK2$   J=J+1 $,
COMMENT CHECK TO SEE IF WEIGHT IS LEGAL $,
          IF J GRT 5 THEN G TO START1 $,
          IF CHAR EQL WAIT(J) THEN GOTO SETWT ELSE GOTO WCHCK2 $,
SETWT$    WT=WAIT(J) $,
          GOTO START2 $,
NULL$     N=0 $,
          WT=0 $,
FINISH$   N=N .LS. 18 $,
          WT=WT+N $,
          RETURN(WT) $,  ... RETURN WEIGHT AND LENGTH OF TERM//
ENC $,
DEFINE PROCEDURE WCOUNT(PT) WHERE INTEGER PT TCBE
COMMENT THIS PROCEDURE COUNTS THE USAGE IN THE SUBJECT INDEX OF
ALL WORDS IN THE TA ARRAY.  THE PROCEDURE PUTS ALL USAGE DATA
IN TO A ARRAY. $,
BEGIN
          INTEGER PTSI,PTC,SL,PTR,PTX,PTXX,PTS,PHRNUM,PHRLEN $,
          INTEGER WT,IL,S,WLEN,P,PHR,BLANK$,PASK $,
          INTEGER PROCEDURE WCHECK,WLNGHT $,
          INTEGER COMPONENT W $,
          SWITCH S:SW1,SW2 $,
          W $=$ L $,
          PASK=77777FC $,
          BLANK=W(4000400040CC $,
          PTSI=W(PT+1) $,
          PTC=W(PT+2) $,
          PTR=W(PT) $,
          IL=W(PT+3) $,  ... GET LENGTH OF TITLE SECTION//
          SL=W(PT+5)+W(PT+6) $,  ... GET FULL LENGTH OF TA//
          PHRNUM=W(PTSI) $,  ... GET NUMBER OF PHRASES//
          S=1 $,  ... SET FLAG TO INDICATE TITLE SECTION//
          P=0 $,
          WLEN=0 $,
NEWREC$   WLEN=WLEN+P $,
          IF WLEN GEQ SL THEN GOTO RETURN $,
          IF WLEN GEQ IL THEN S=2 $,  ... WHEN ABSTRACT SECTION REACHED, RESET FLAG//
          PTR=PTR+P $,  ... SET POINTER TO NEXT WORD OF TA//
          PTC=PTC+3 $,
          PTS=PTSI+1 $,  ... SET POINTER TO NEXT PHRASE//
          W(PTSI)=C $,
          W(PTS)=0 $,  ... INITIALIZE DATA CELLS FOR CURRENT >ERC//
          W(PTS+1)=0 $,
          P=W(PTR) $,
          P=W.LNGTH(PTR) $,  ... GET LENGTH OF CURRENT WORC//
          P=P+0 $,
NEWPHR$   PTS=PTS+PHRLEN $,
COMMENT SEARCH ALL PHRASES FOR CURRENT WORD USAGES $,
          IF PHR GEQ PHRNUM THEN GOTO STOR $,
          PHR=PHR+1 $,
COMMENT GET LENGTH OF PHRASE AND WEIGHT $,
          PHRLEN=W(PTSI) .RS. 18 $,
          WT=W(PTSI) .A. 777C $,
          PTS=PTS+1 $,
COMMENT INITIALIZE FOR CALL TO WCHECK $,
          W(PT+5)=PTS+PHRLEN .LS. 18 $,
          W(PT+11)=PTS+14+N .LS. 18 $,
          PTR=WCHECK(PT) $,  ... SEE IF CURRENT WORC IS USED IN PHRASE//
          LEN1=LEN-4 $,
          LEN2=LEN .LS. 18 $,
          PTR=PTR-LEN2 $,
          PTX=PTX+LEN1 .LS. 18 $,
          WT=WT .A. MASK $,
COMMENT KEEP TRACT OF THE NUMBER OF OCCURRENCES OF EACH WEIGHT
SUBJECT TERM IN THE BOOK FILE $,
          IF WT EQL 060C THEN W(BKPT+2)=W(BKPT+2)+1CCOCCCCCC $,
          IF WT EQL 061C THEN W(BKPT+2)=W(BKPT+2)+1CCOOC $,
          IF WT EQL 062C THEN W(BKPT+2)=W(BKPT+2)+1 $,
          IF WT EQL 063C THEN W(BKPT+3)=W(BKPT+3)+1CCOOC $,
          IF WT EQL 064C THEN W(BKPT+3)=W(BKPT+3)+1 $,
          INC(PTR,LEN) $,  ... MOVE PTR TO START OF NEXT TERM//
          W(PT+8)=PTX $,
          W(PT+5)=PTS+1 $,
          SIL=TRANFER(PT) $,  ... MOVE THE TERM TC SI//
          RL=RL+SIL+1 $,  ... INCREMENT THE FILLED LENGTH CF SI//
          IF RL GRT 880 THEN GOTO FINISH $,
COMMENT IF SI GETS TOO FULL THEN ABORT $,
          SIL=SIL .LS. 18 $,
          W(PTSI)=SIL+WT $,  ... PUT IN TERM HEADER WITH WEIGHT AND LENGTH//
          PTS=PTS+SIL+1 $,
          PHR=PHR+1 $,  ... INCREASE NUMBER CF TERMS//
          IF ILEN GRT LENLIM THEN GOTO FINISH $,
          GOTO START $,
FINISH$   W(PTSI)=PHR $,  ... PUT IN SI HEADER//
          TRNS(=RL $,
ENC $,
DEFINE INTEGER PROCEDURE WEIGHT(PTR) WHERE INTEGER PTR TCBE
COMMENT THIS PROCEDURE FINDS THE WEIGHT OF A SUBJECT TERM IN
THE SUBJECT INDEX FIELD.  PTR IS A STANDARD INTREX POINTER TC
THE START CF THE TERM.  THE PROCEDURE RETURNS BOTH THE WEIGHT AND
LENGTH OF THE TERM. $,
BEGIN
          INTEGER PTX,NUM,I,J,N,WT,CHAR $,
          INTEGER ARRAY CLIST(5),WAIT(6) $,
          INTEGER PROCEDURE GET $,
          PROCEDURE INC $,
          PRESET BEGIN
COMMENT SET THE LIST OF DELIMITERS FOR WEIGHTS AND THE POSSIBLE WEIGHTS $,
            CLIST(1)=050C $,
            CLIST(3)=051C $,
            CLIST(4)=073C $,
            WAIT(1)=060C $,
            WAIT(2)=061C $,
            WAIT(3)=062C $,
            WAIT(4)=063C $,
            WAIT(5)=064C
          END $,
          NUM=PTR .RS. 18 $,
          NUM=NUM .A. 77777C $,
          PTX=PTR $,
          N=0 $,
START1$   I=0 $,
START2$   I=I+1 $,
COMMENT SEARCH FOR THE PROPER DELIMITERS $,
          IF I GRT 4 THEN GOTO FINISH $,
          CHAR=GET(PTX) $,
          INC(PTR,1) $,
          N=N+1 $,
          IF N GRT NUM THEN GOTO NULL $,
```

```
          IF PTR EQL 0  THEN GOTO NXPHR $,
          GOTO $(6) $,
COMMENT IF WORD IS FOUND IN TITLE THEN UPDATE DATA STORED IN ARRAY D $,
$W16      IF W( EQL 0A0C THEN W(PTD-2)=W(PTC-2)+4CCCCCCCC00C $,
          IF W( EQL 0A1C THEN W(PTD-2)=W(PTC-2)+1 $,
          IF W( EQL 0A2C THEN W(PTD-2)=W(PTC-2)+4CCCC00C $,
          IF W( EQL 0A3C THEN W(PTD)=W(PTD)+100CCCCCC3C $,
          IF W( EQL 0A4C THEN W(PTD-2)=W(PTC-2)+2CC0C $,
          GOTO NXPHR $,  ... GO TO THE NEXT PHRASE//
COMMENT IF THE WORD IS FROM THE ABSTRACT THEN UPDATE ARRAY D $,
$W28      IF W( EQL 0A0C THEN W(PTD-1)=W(PTC-1)+4CCCCC00D0C $,
          IF W( EQL 0A1C THEN W(PTD-1)=W(PTC-1)+1 $,
          IF W( EQL 0A2C THEN W(PTD-1)=W(PTC-1)+4CCCCC00C $,
          IF W( EQL 0A3C THEN W(PTD)=W(PTD)+1000C $,
          IF W( EQL 0A4C THEN W(PTD-1)=W(PTC-1)+2C00C $,
          GOTO NXPHR $,
STCR8     IF K EQL 1 THEN W(PTC)=W(PTU)+1CCCC00C ELSE W(PTC)=W(PTC)+1 $,
COMMENT RECORD THE APPEARANCE OF THE WORD AS EITHER TITLE OR
ABSTRACT APPEARANCE. $,
          GOTO SAVRD $,
END $,
END FINI
```

```
* ********************************************************************************
     M5806       CMPL01        FMAN        ALGOL FOR   M5806        4976      051548
BEGIN
DEFINE PROCEDURE FSRCH(PTR,PT) WHERE INTEGER PTR,PT TOBE
COMMENT PTR POINTS TO A WORD TO BE SEARCHED FOR IN THE ACTIVE
DICTIONARY FILE.  IF THE WORD IS FOUND, THEN THE APPROPRIATE
DATA WILL BE ACCES TO THE DICTIONARY FILE. IF THE WORD IS NOT
FOUND THEN THE WORD WILL BE ADDED TO THE FILE. $,
BEGIN
          INTEGER M,DCTPT,BKPT,FN,WN,P1,COUNT,PTC,PTRR,I $,
          PROCEDURE RPLACE $,
          INTEGER COMPONENT W $,
          M $=$ C $,
          M=W(PT+5) $,  ... GET WORD LENGTH FOR FILE//
          BKPT=W(PT+12) $,
          DCTPT=W(PT+13) $, ... GET START LOC OF DICTIONARY FILE//
          PTD=W(PT+2)+W(PT+11) $,  ... GET LOC OF WORD DATA//
          FN=W(BKPT+2*M) $,
COMMENT GET MAX FILE NUMBER FROM THE BOOK FILE. ALSO
GET NUMBER OF LOCATIONS USED IN THE LAST FILE. $,
          WN=W(BKPT+2*M+1) $,
          P1=P-1 $,
          COUNT=0 $,
START$    I=0 $,
          IF W(PT+10) EQL FN THEN GOTO CHECK $,
COMMENT W(PT+10) CONTAINS THE NUMBER OF THE CURRENT FILE $,
          IF COUNT GRT 862-P THEN GOTO RETURN $,
COMMENT IF THERE ARE STILL MORE WORDS IN THE FILE THEN CONTINUE $,
LOOP1$    IF W(PT+1) NEQ W(DCTPT+COUNT+I) THEN GOTO CONT $,
COMMENT CHECK THE PRESENT WORD TO SEE IF IT MATCHES THE WORD
POINTED TO BY PTR $,
          I=I+1 $,
          IF I LES M1 THEN GOTO LOOP1 $,
COMMENT IF THE WORD IS FOUND THEN TRANSMIT DATA FROM ARRAY D
TO THE DICTIONARY FILE $,
          W(DCTPT+COUNT+P1)=W(DCTPT+CCUNT+M1)+W(PTD-2) $,
          W(DCTPT+COUNT+M1+1)=W(DCTPT+COUNT+P1+1)+W(PTD-1) $,
          W(DCTPT+COUNT+M1+2)=W(DCTPT+COUNT+P1+2)+W(PTD) $,
          PTRR=PTR+(M .LS. 18) $,
          RPLACE(PTRR) $,
COMMENT RPLACE THE WORD IN TA WITH A SPECIAL MARK TO SHOW IT HAS
BEEN FOUND AND THE DATA TRANSMITTED $,
          GOTO RETURN $,
CONT$     CCUNT=CCUNT+M+2 $,
          GOTO START $,
CHECK$    IF COUNT LES WN THEN GOTO LOOP1 $,
COMMENT IF THE END OF THE FILE IS EXCEEDED THEN RETURN $,
          IF COUNT GRT 862-P THEN GOTO RETURN $,
COMMENT IF THERE IS ROOM FOR A NEW WORD IN THE FILE, THEN ADD IT $,
          I=0 $,
STOR$     W(DCTPT+COUNT+I)=W(PTR+I) $,
          I=I+1 $,
          IF I LES M1 THEN GOTO STOR $,
          W(DCTPT+COUNT+M1)=W(PTD-2) $,
          W(DCTPT+COUNT+M1+1)=W(PTD-1) $,
          W(DCTPT+COUNT+M1+2)=W(PTD) $,
          W(BKPT+2*M+1)=W(BKPT+2*M+1)+M+2 $,
          PTRR=PTR+(M .LS. 18) $,
          RPLACE(PTRR) $,
END $,
DEFINE PROCEDURE CCTFIL(TYPE,ACT,PT) WHERE INTEGER TYPE, ACT,PT TORE
```

```
COMMENT THIS PROCEDURE MERELY OPENS THE SPECIFIED DICTIONARY
FILE.  ACT MAY BE EITHER OPEN OR CLOSE.  THE FILE WORD LENGTH MUST
BE CONTAINED IN PTRS(9) AND THE FILE NUMBER MUST BE IN PTRS(10).
TYPE IS IGNORED AND MAY BE ANYTHING $,
BEGIN
          INTEGER DICT,BKPT,M,OPN,CLOS,N,N1,N2,NM1,NM2,RW,ECF,EOFCT $,
          INTEGER ARRAY CF(864) $,
          PROCEDURE OPEN,ROWAIT,WRWAIT,CLOSE $,
          PROCEDURE BUFFER,FSTATE $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          PRESET BEGIN
            DICT .BCD. /  DICT/ $,
            OPN  .BCD. /  OPEN/ $,
            CLOS .BCD. / CLOSE/ $,
            NM1=3333334500000 $,
COMMENT PRESENT FILE NAME TO ...N00 ...N00 $,
            NM2=3333334400000 $,
            RW .BCD. /    RW/ $,
          END $,
          W(PT+13)=LOC OF $,
COMMENT PUT STARTING LOC OF THE DICTIONARY FILE IN PTRS(13) $,
          BKPT=W(FT+12) $,
          IF ACT EQL OPN THEN BEGIN
            M=W(PT+9) $,
            N=W(PT+10) $,
            IF N CEQ 10 THEN N=(N/10)*66C+W(PT+10) $,
COMMENT GENERATE ECD NAMES OF FILE $,
            N1=NM1+N $,
            N2=NM2+M $,
            OPEN(RW,N1,N2,0,2) $,
            IF W(FT+10) GRT W(BKPT+2*M) THEN BEGIN
COMMENT IF THE FILE IS NEW, INCREMENT THE ENTRY IN BOOK FILE $,
              W(BKPT+2*M+1)=C $,
              W(BKPT+2*M)=W(BKPT+2*M)+1 $,
              GOTO RETURN $,
            END $,
            ROWAIT(N1,N2,0,CF TO 864,ECF,EOFCT) $,
          END $,
          IF ACT EQL CLOS THEN BEGIN
            WRWAIT(N1,N2,1,CF TO 864,EOF,EOFCT) $,
            CLOSE(N1,N2) $,
            W(PT+10)=W(PT+10)+1 $,  ... INCREMENT FILE NUMBER FOR NEXT CALL//
          END $,
END $,
END FINI
```

```
      M5806      CMFL01          WC       ALGOL FOR   M5806          4474
BEGIN
DEFINE INTEGER PROCEDURE WCHECK(PTR) WHERE INTEGER PTR TOO$
COMMENT SEARCH AN ARRAY POINTED TO BY THE CONTENTS OF
PTRS(5) TO SEE IF IT CONTAINS THE WORD POINTED TO BY THE
CONTENTS OF PTRS(11). $,
BEGIN
          INTEGER M,M1,I,WRC,PTX,WPT,XL,PTENC $,
          INTEGER PROCEDURE WLNGTH $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          PTX=W(PT+5) .A. 77777C $,  ... GET PTR TO ARRAY//
          XL=W(PT+5) .RS. 18 $,  ... GET LENGTH OF ARRAY//
          WPT=W(FT+11) .A. 77777C $,  ... GET PTR TO WORD//
          M=W(PT+11) .RS. 18 $,  ... GET WORD LENGTH//
          M1=0 $,
          WRD=0 $,
          PTENC=PTX+XL $,
START$    PTX=PTX+M1 $,  ... FIND NEXT WORD OF ARRAY//
          WRD=WRC+3 $,
          IF PTX CEQ PTENC THEN GOTO FINISH $,
          M1=WLNGTH(PTX) $,
          IF M1 EQL M THEN BEGIN
COMMENT IF WORD LENGTHS MATCH THEN SEE IF WORDS ARE SAME *,
            I=0 $,
  LOOP$     IF W(BPT+I) NEQ W(PTX+I) THEN GOTO START $,
            I=I+1 $,
            IF I LES M THEN GOTO LOOP $,
            GOTO CONST $,
          END $,
          GOTO START $,
FINISH$   PTX=0 $,  ... IF WORD NOT FOUND IN ARRAY//
          WRD=0 $,  ... THEN RETURN A 0//
COMMENT ELSE RETURN A POINTER TO THE POSITION OF THE WORD
IN THE ARRAY AND IN THE DECREMENT THE NUMBER OF THE WORD+1 $,
CONST$    WCHECK=PTX+(WRD .LS. 18) $,
END $,
END FINI
```

The programs of Phase II are very similar to those of Phase I. The structure of IA and SI are nearly identical to the same arrays of Phase I. The dictionary used as a reference for Phase II is just that generated in Phase I. The major difference in the programming for Phase II is first, that rather than place data into the dictionary, Phase II extracts data from the dictionary. Second, Phase II must operate on the extracted data to form a pre-index. Thus, changes have been made to the D array; also, two extra arrays, P and PI, have been provided to help in generating pre-indexes. Thus the directory array, PTRS, for Phase II contains two additional entries. PTRS(14) contains a pointer to P; PTRS(15) a pointer to PI.

The D Array

The D array is still used to store the usage data for all words in TA. The usage data in D can than serve as an immediate reference for evaluation of pre-indexes. However, in Phase II only one computer word for D is used to store usage information, rather than the three computer words used in Phase I.

The P Array

The data extracted from the dictionary for all words of are placed in array P. One computer word in P is used for each word of TA. The data placed in P are the usage rate for the word and the frequency of appearance of the word.

The PI Array

The PI array is used for flags which mark those words selected for a pre-index. The rightmost nine bits of a word in PI

are used to indicate selected words. Three flags are reserved for each method in order to handle three parameter sets simultaneously.

The PARA Array

One further set of data is necessary to operate Phase II. Three sets of parameters must be specified for each pre-indexing method. PARA holds 18 parameters in total. Method I requires three parameters for three trials; method II, six; and method III, nine.

Operation of Phase II

To operate Phase II, all Phase II programs must be loaded and started from a CTSS console. The files CATDIR FILE and CATREC FILE must be available. The program will request the user to input the parameter sets for each method. The parameters must be specified as two-digit integers on a scale from 1 to 31. The program converts the parameters to appropriate usage rate for frequency thresholds.

Once the parameters have been typed in, the program will request a document number. The user may type a four-digit number. When the pre-indexes have been run, the program will printout a summary evaluation and wait for the next document number.

To obtain a full listing of the pre-indexes for a document, the user may type a 1 after the document number. The pre-indexing system will print a table showing each word of the title and abstract. and show whether the word has been included in a pre-index for each method and each parameter set.

```
*************************************************************************************
     M5806      CMFL01       PMNN2       ALGOL FOR   M58C6        4976       951568
BEGIN
INTEGER ARRAY TA(1200),SI(900),D(300),P(300),PI(300) $,
INTEGER ARRAY BK(432),PTRS(17),PARA(16) $,
INTEGER PT,EOF,ECFCT,I,BOOK,FILE,RW $,
INTEGER N,Z,BKPT,B $,
PROCEDURE SYNCOM,CARET,IGTFLD,OPEN,CLOSE,PRT,ASMBCD $,
PROCEDURE WCNT2,EVAL,DCT2,RDWAIT $,
INTEGER PROCEDURE SET2 $,
INTEGER COMPONENT W $,
W $=$ 0 $,
PRESET BEGIN
BCCK .BCD. /  BCCK/ $,
FILE .BCD. /  FILE/ $,
RW .BCD. /R/ $,
END $,
COMMENT INITIALIZE THE PTRS ARRAY WITH ALL DIRECTORY INFORMATION $,
PT=LOC PTRS $,
PTRS(0)=LOC TA(1) $,
PTRS(1)=LOC D $,
PTRS(2)=LOC BK $,
PTRS(4)=LOC P $,
PTRS(5)=LOC PI $,
B=LOC PARA $,
CARET() $,
PRT(ASMBCD,,C. /TYPE PARAMETERS FOR METHOD 1/) $,
PRT(ASMBCD,,C. /R* R* R*/) $,
READ FP1,PARA(0),PARA(1),PARA(2) $,
COMMENT READ IN PARAMETERS FOR METHOD 1 $,
CARET() $,
PRT(ASMBCD,,C. /TYPE PARAMETERS FOR METHOD 2/) $,
PRT(ASMBCD,,C. /R* F* R* F* R* F*/) $,
READ FP2,PARA(3),PARA(6),PARA(4),PARA(7),PARA(5),PARA(8) $,
COMMENT READ IN PARAMETERS FOR METHOD 2 $,
CARET() $,
PRT(ASMBCD,,C. /TYPE PARAMETERS FOR METHOD 3/) $,
PRT(ASMBCD,,C. /R1 R2 F* R1 R2 F* R1 R2 F*/) $,
READ FP3,PARA(9) TO PARA(17) $,
COMMENT READ IN PARAMETERS FOR METHOD 3 $,
CARET() $,
FP1$ FORMAT(I2,2I3) $,
FP2$ FORMAT(I2,5I3) $,
FP3$ FORMAT(I2,8I3) $,
IGTFLD() $,  ... INITIALIZE CATALOG FILES//
OPEN(RW,BOOK,FILE,0,2) $,  ... GET DIRECTORY FILE FOR DICTIONARY//
RDWAIT(BOOK,FILE,C,BK TO 432,EOF,EOFCT) $,
CARET() $,
CARET() $,
PRINT FM $,
FM$ FORMAT(42X,8HMETHOD 1,8X,8HMETHOD 2,8X,8HMETHOD 3) $,
PRT(ASMBCD,,C. /TYPE DOCUMENT NUMBERS/) $,
START$ READ F2,PTRS(6),PTRS(16) $,
COMMENT READ NUMBER OF DOCUMENT TO BE PRE-INDEXED. PTRS(16)
WILL BE 1 IF FULL PRE-INDEX IS TO BE PRINTED, 0 IF SUMMARY ONLY $,
F2$ FORMAT(I4,I2) $,
IF PTRS(6) EQL 0 THEN GOTO FINISH $,
COMMENT IF DOCUMENT NUMBER IS 0 THEN QUIT $,
Z=SET2(PT) $,  ... SET UP ARRAYS TA AND SI//
IF Z GRT 0 THEN BEGIN
COMMENT IF A FEILD IS MISSING THEN PRINT FEILD NUMBER
END START OVER $,
PRINT F,PTRS(6),Z $,
F$ FORMAT(4H DOC,I4,14H MISSING FIELD,I3) $,
GOTO START $,
END $,
WCNT2(PT) $,  ... ANALYZE ACTUAL SUBJECT INDEX//
DCT2(PT) $,  ... OBTAIN DICTIONARY DATA ON ALL WORDS OF TA//
EVAL(PT,B) $,  ... GENERATE A PRE-INDEX AND COMPARE TO ACTUALS//
CARET() $,
GOTO START $,
FINISH $ CLOSE(BCCK,FILE) $,
CHAOSM() $,
END FINI
```

```
      M5806      CMFL01         SEC2        ALGOL FOR   M58C6         4976       0515C8
BEGIN
DEFINE PROCEDURE CCT2(PT) WHERE INTEGER PT TCBE
BEGIN
COMMENT THIS PROCEDURE SEARCHES THE DICTIONARY FILES TO FIND
ALL THE WORDS OF THE TA ARRAY.  WHEN A WORD IS FOUND, USAGE
RATE AND FREQUENCY DATA ARE TRANSMITTED TO ARRAY P BY OTHER
ROUTINES.  IF A WORD IS NOT FOUND, THE OTHER ROUTINES INSERT
A -1 IN THE APPROPRIATE ENTRY IN ARRAY P. $,
          INTEGER WRD1,WRD2,PTP,M,M1,M2,OPEN,CLOSE,PTENC,PTX,PTXX $,
          INTEGER PROCEDURE WLNGTH $,
          PROCEDURE DF2,FS2 $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          PRESET BEGIN
          OPEN .ECO. /  OPEN/ $,
          CLOSE .ECO. / CLOSE/ $,
          END $,
          PTENC=W(PT)+W(PT+2)+W(PT+4) $,  ... LOCATE END OF TA//
          PTP=W(PT+14) $,
START0$   W(PT+1C)=1 $,  ... SET THE FILE NUMBER TO 1//
          P=0 $,
          WRD1=C $,
          PTX=W(PT) $,
START1$   PTX=PTX+M $,  ... POINT TO NEXT WORD OF TA//
          WRD1=WRC1+1 $,
          IF PTX CEQ PTENC THEN BEGIN
           W(PT+11)=WRD1 $,  ... RETURN NUMBER OF WORDS IN TA//
           W(PTP+WRD1)=0 $,
           GOTO RETURN $,
          END $,  ... GET LENGTH OF NEXT WORC//
          M=WLNGTH(PTX) $,
          IF W(PTF+WRD1) NEC 0 THEN GCTO START1 $,
          W(PT+S1+M $,  ... OPEN THE PROPER DICTIONARY FILE//
START2$   CF2(OPEN,PT) $,  ... SET WORD NUMBER POINTED AT//
FS$       W(PT+12)=WRD1 $,  ... SEARCH FOR WORD IN DICTIONARY FILE//
COMMENT IF WORD FOUND IN DICTIONARY THEN SEARCH FOR NEXT
WORC CF TA THAT HAS SAME LENGTH $,
          FS2(PT),PT) $,
          IF W(PTF+WRD1) NEC 0 THEN BEGIN
             M1=M $,
NWRC$        PTX=PTX+M1 $,
             WRC1=WRD1+1 $,
             IF PTX GEQ PTENC THEN BEGIN
               CF2(CLOSE,PT) $,
               GCTU START0 $,
             ENC $,
             M1=WLNGTH(PTX) $,
             IF W(PTP+WRD1) NEQ 0 THEN GOTO NWRD $,
             IF M1 EQL M THEN GCTO FS $,
             GOTC NWRD $,
          END $,
          PTXX=PTX $,
          WRD2=WRC1 $,
          M2=M $,
COMMENT IF WORD IS NOT FOUNC IN THE DICTIONARY FILE THEN SEARCH
FOR THE NEXT WORC OF THE SAME LENGTH $,
LCCP$     PTXX=PTXX+M2 $,
          WRD2=WRC2+1 $,
          IF PTXX GEQ PTEND THEN BEGIN
             DF2(CLOSE,PT) $,
             GOTC START2 $,
          END $,
          M2=WLNGTH(PTXX) $,
          IF W(PTF+WRD2) NEC 0 THEN GOTO LCCP $,
          IF M2 ECL M THEN BEGIN
             W(PT+11)=WRD2 $,
             FS2(PTXX,PT) $,
          END $,
          GCTO LCCP $,
ENC $,
DEFIN: INTECER PRCCEDURE SET2(PT) WHERE INTEGER PT TCBE
COMMENT THIS PRCCEDURE OBTAINS THE TITLE, ABSTRACT AND SUBJECT
INCEX OF A COCUMENT AND PLACES THEM IN THE PROPER MANNER
IN THE PROPER ARRAYS. $,
BEGIN
          INTEGER PROCEDURE GETFLD,TRNSI,TRNFER $,
          INTEGER DN,FN,YL,AL,I $,
          INTEGER ARRAY XN(1C) $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          XN=9 $,
          CN=W(PT+6) $,  ... GET DOCUMENT NUMBER//
          FN=24 $,  ... SET FIELD NUMBER FOR TITLE//
          W(PT+5)=W(PT) $,  ... SPECIFY STARING LOCATION FCR FIELD//
          W(PT+8)=GETFLD(DN,FN,XN) $,
          IF W(PT+8) LES 0 THEN GOTO FINISH $,
COMMENT IF FIELC NOT FOUNC THEN GOTO RETURN $,
          W(PT+3)=TRNFER(PT) $,
COMMENT MOVE BUFFER AND SET LENGTH OF TITLE SECTICN CF TA.$,
          FN=71 $,
          W(PT+5)=W(PT)+W(PT+3) $,  ... SPECIFY WHERE ABSTRACT TC START//
          W(PT+8)=GETFLD(DN,FN,XN) $,
          IF W(PT+8) LES 0 THEN GOTO FINISH $,
          W(PT+4)=TRNFER(PT) $,  ... SET LENGTH CF ABSTRACT SECTICN IN TA//
          IF W(PT+4) EQL 0 THEN GOTO FINISH $,
FN73$     FN=73 $, ... SET FIELC TO SUBJECT INDEX//
          W(PT+5)=W(PT+1) $,  ... SPECIFY START LOC FOR SI ARRAY//
          W(PT+8)=GETFLD(DN,FN,XN) $,
          IF W(PT+8) LES 0 THEN GOTO FINISH $,
          I=TRNSI(PT) $,  ... MCVE BUFFER TC SI, RETURN LENGTH CF SI//
COMMENT IF SI TCC LONG THEN QUIT  $,
          IF I GRT 880 THEN GOTO FINISH $,
          FN=0 $,  ... IF ALL FIELDS MCVED PROPERLY, RETURN C//
FINISH$   SET2=FN $,
ENC $,
ENC FINI
```

```
**************************************************************************
      M5806        CMFL01          TRN2         ALGOL FOR    M5806          4976        051568
BEGIN
DEFINE INTEGER PROCEDURE WLNGTH(PTT) WHERE INTEGER PTT TOBE
BEGIN
COMMENT THIS PROCEDURE IS IDENTICAL TO WLNGTH OF PHASE I $,
            INTEGER M $,
            INTEGER COMPONENT W $,
            W $=$ C $,
            M=0 $,
 LOOP1$     IF W(PTT+M) EQL 04C04C04004CC THEN GOTO LOOP2 $,
            M=M+1 $,
            GOTO LOOP1 $,
LOOP2$      M=M+1 $,
            IF W(PTT+M) EQL 04C04C04004CC THEN GOTO LOOP2 $,
            WLNGTH=M $,
END $,
DEFINE INTEGER PROCEDURE TRNFER(PT) WHERE INTEGER PT TOBE
BEGIN
COMMENT THIS PROCEDURE IS IDENTICAL TO TRNFER OF PHASE I EXCEPT
THAT NO CALL IS MADE TO WCHECK TO GUARD AGAINST MULTIPLE APPEARANCES
OF WORDS IN THE WORD STRING THAT IS TRANSFERED $,
            INTEGER I,K,M,L,XL,CK,PTR,PTX,PTXX $,
            INTEGER PTTEMP,BLANKW,BLANKS,CHAR,NUM $,
            INTEGER ARRAY TEMP(9),BRLIST(30) $,
            INTEGER PROCEDURE GET $,
            PROCEDURE INC $,
            INTEGER COMPONENT W $,
            SWITCH S=SW1,SW2 $,
            W $=$ C $,
            BLANKW=C4004004004CC $,
            PTTEMP=LOC TEMP(1) $,
            XL=0 $,
            PTR=W(PT+8) $,
            PTX=W(PT+5) $,
            NUM=PTR .RS. 18 $,
            NUM=NUM .A. 77777C $,
START$      L=27 $,
            M=1 $,
            K=1 $,
            FOR I=1,2,3,4,5,6,7,8 DO TEMP(I)=C $,
START1$     IF NUM LEQ 0 THEN GOTO FINISH $,
            CHAR=GET(PTR) $,
            INC(PTR,1) $,
            NUM=NUM-1 $,
            GOTO S(K) $,
SW1$        IF CHAR GEQ 101C THEN BEGIN
              IF CHAR LEQ 140C THEN CHAR=CHAR+C40C $,
              IF CHAR LEQ 172C THEN GOTO BUILD $,
             GOTO FINISH $,
            END $,
            IF CHAR LEQ 051C THEN GOTO FINISH $,
            IF CHAR GEQ 055C THEN GOTO FINISH $,
            IF CHAR GEQ 053C THEN GOTO FINISH $,
            IF CHAR EQL 052C THEN BEGIN
              K=2 $,
              GOTO BUILD $,
            END $,
SW2$        IF CHAR EQL 052C THEN K=1 $,
BUILD$      CHAR=CHAR .LS. L $,
            TEMP(M)=TEMP(M)+CHAR $,
            IF L EQL 0 THEN BEGIN
              L=27 $,
              M=M+1 $,
              IF M GRT 8 THEN BEGIN
                PRINT F1,W(PT+6) $,
F1$             FORMAT(17H LONG WORD IN DOC,I5) $,
                GOTO START $,
              END $,
              GOTO START1 $,
            END $,
            L=L-9 $,
            GOTO START1 $,
FINISH$     IF L NEQ 27 THEN GOTO CONT $,
            IF M EQL 1 THEN BEGIN
              IF NUM GRT 0 THEN GOTO START ELSE GOTO DONE $,
            END $,
            GOTO BYPASS $,
CONT$       L=27-L $,
            BLANKS=BLANKW .RS. L $,
            TEMP(M)=TEMP(K)+BLANKS $,
            M=M+1 $,
BYPASS$     TEMP(M)=BLANKW $,
            PTXX=PTX+XL $,
            I=0 $,
STOR$       IF I EQL M THEN BEGIN
              XL=XL+M $,
            IF XL GRT 1150 THEN BEGIN
              PRINT F2,W(PT+6) $,
F2$           FORMAT(19H FIELD TOO LONG DOC,I5) $,
              XL=C $,
              GOTO DONE $,
            END $,
              IF NUM LEQ 0 THEN GOTO DONE $,
              GOTO START $,
            END $,
            W(PTXX+I)=TEMP(I+1) $,
            I=I+1 $,
            GOTO STOR $,
DONE$       TRNFER=XL $,
END $,
DEFINE INTEGER PROCEDURE TRNSI(PT) WHERE INTEGER PT TOBE
COMMENT THIS PROCEDURE IS IDENTICAL TO TRNSI OF PHASE I $,
BEGIN
            INTEGER PTR,BKPT,XL,PTSI,PMR,PTX,LEN,WT,$IL,$IL1,LOW1,PTS
            INTEGER MASK,LEN2,TLEN,LENLIM $,
            INTEGER PROCEDURE WEIGHT,TRNFER $,
            PROCEDURE INC $,
            INTEGER COMPONENT W $,
            W $=$ C $,
            MASK=777777C $,
            XL=1 $,
            TLEN=0 $,
            PMR=0 $,
            PTR=W(PT+8) $,
            PTSI=W(PT+11) $,
            BKPT=W(PT+12) $,
            LENLIM=PTR .RS. 18 $,
            LENLIM=LENLIM .A. 77777C $,
            PTS=PTSI+1 $,
START$      WT=WEIGHT(PTR) $,
            IF WT EQL 0 THEN GOTO FINISH $,
```

```
          LEN=WT .RS. 18 $,
          TLEN=TLEN+LEN $,
          PTX=PTR .A. MASK $,
          LEN1=LEN-4 $,
          LEN2=LEN .LS. 18 $,
          PTR=PTR-LEN2 $,
          PTX=PTX+LEN1 .LS. 18 $,
          WT=WT .A. MASK $,
          IF WT EQL 060C THEN W(BKPT+2)=W(BKPT+2)+1CCCCCCCCC $,
          IF WT EQL 061C THEN W(BKPT+2)=W(BKPT+2)+1CCCCC $,
          IF WT EQL 062C THEN W(BKPT+2)=W(BKPT+2)+1 $,
          IF WT EQL 063C THEN W(BKPT+3)=W(BKPT+3)+1CCCCC $,
          IF WT EQL 064C THEN W(BKPT+3)=W(BKPT+3)+1 $,
          INC(PTR,LEN) $,
          W(PT+8)=PTX $,
          W(PT+5)=PTS+1 $,
          SIL=TRNFER(PT) $,
          XL=XL+SIL+1 $,
          IF XL GRT 880 THEN GOTO FINISH $,
          SIL1=SIL .LS. 18 $,
          W(PTS)=SIL1+WT $,
          PTS=PTS+SIL+1 $,
          PHR=PHR+1 $,
          IF TLEN GRT LENLIM THEN GOTO FINISH $,
          GOTO START $,
FINISH$   W(PTS1)=PHR $,
          TRNS1=XL $,
END $,
DEFINE INTEGER PROCEDURE WEIGHT(PTR) WHERE INTEGER PTR TOBE
BEGIN
COMMENT THIS PROCEDURE IS IDENTICAL TO WEIGHT OF PHASE 1 $,
          INTEGER PTX,NUM,I,J,N,WT,CHAR $,
          INTEGER ARRAY CLIST(5),WAIT(6) $,
          INTEGER PROCEDURE GET $,
          PROCEDURE INC $,
          PRESET BEGIN
            CLIST(1)=050C $,
            CLIST(2)=051C $,
            CLIST(4)=073C $,
            WAIT(1)=060C $,
            WAIT(2)=061C $,
            WAIT(3)=062C $,
            WAIT(4)=063C $,
            WAIT(5)=064C
          END $,
          NUM=PTR .RS. 18 $,
          NUM=NUM .A. 77777C $,
          PTX=PTR $,
          A=0 $,
START.$   I=0 $,
START2$   I=I+1 $,
          IF I GRT 4 THEN GOTO FINISH $,
          CHAR=GET(PTX) $,
          INC(PTX,1) $,
          A=A+1 $,
          IF A GRT NUM THEN GOTO NULL $,
          IF I EQL 2 THEN GOTO WCHCK1 $,
          IF CHAR NEQ CLIST(I) THEN GOTO START1 ELSE GOTO START2 $,
WCHCK.$   J=0 $,
WCHCK1$   J=J+1 $,
          IF J GRT 5 THEN GOTO START1 $,
          IF CHAR EQL WAIT(J) THEN GOTO SETWT ELSE GOTO WCHCK2 $,
SETWT$    WT=WAIT(J) $,
          GOTO START2 $,
NULL$     A=0 $,
          WT=0 $,
FINISH$   A=A .LS. 18 $,
          WT=WT+A $,
          WEIGHT=WT $,
END $,
DEFINE PROCEDURE WCNT2(PT) WHERE INTEGER PT TOBE
BEGIN
COMMENT THIS PROCEDURE IS NEARLY IDENTICAL TO WCOUNT OF PHASE 1.
THE ONLY DIFFERENCE IS IN THE WAY THAT USAGE DATA IS PLACED IN
ARRAY D. IN WCNT2 ALL USAGE DATA IS PLACED IN ONE COMPUTER WORD
RATHER THAN 3.  ALSO NO APPEARANCE DATA IS RECORDED SINCE EACH
WORD HAS EXACTLY ONE APPEARANCE FOR EACH OCCURRENCE. $,
          INTEGER PTS1,PTD,XL,PTR,PTX,PTXX,PTS,PHRNUP,PHRLEN $,
          INTEGER WT,TL,K,WLEN,M,PHR,BLANKW,MASK $,
          INTEGER PROCEDURE WCHECK,WLNGHT $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          MASK=777777C $,
          BLANKW=C4006004004CC $,
          PTS1=W(PT+1) $,
          PTD=W(PT+2) $,
          PTX=W(PT) $,
          XL=W(PT+3)+W(PT+4) $,
          PHRNUM=W(PTS1) $,
          M=0 $,
          WLEN=C $,
NXWRD$    WLEN=WLEN+M $,
          IF WLEN GEQ XL THEN GOTO RETURN $,
          PTX=PTX+M $,
          PTD=PTD+1 $,
          PTS=PTS1+1 $,
          W(PTD)=C $,
          PHRLEN=C $,
          M=WLNGTH(PTX) $,
          PHR=0 $,
NXPHR$    PTS=PTS+PHRLEN $,
          IF PHR GEQ PHRNUM THEN GOTO NXWRD $,
          PHR=PHR+1 $,
          PHRLEN=W(PTS) .RS. 18 $,
          WT=W(PTS) .A. 777C $,
          PTS=PTS+1 $,
          W(PT+5)=PTS+PHRLEN .LS. 18 $,
          W(PT+11)=PTX+M .LS. 18 $,
          PTR=WCHECK(PT) $,
COMMENT IF A USAGE IS FOUND INSERT THE USAGE DATA IN ARRAY D $,
          IF PTR EQL 0 THEN GOTO NXPHR $,
          IF WT EQL 060C THEN W(PTD)=W(PTD)+100CCCCCCC $,
          IF WT EQL 061C THEN W(PTD)=W(PTD)+100CCCCC $,
          IF WT EQL 062C THEN W(PTD)=W(PTD)+100CC $,
          IF WT EQL 063C THEN W(PTD)=W(PTD)+100C $,
          IF WT EQL 064C THEN W(PTD)=W(PTD)+1 $,
          GOTO NXPHR $,
END $,
END FINI
```

```
      P5806       (MFL01          FMN2        ALGOL FOR    P58C6          4976       0515C8
BEGIN
DEFINE PROCEDURE FS2(PTX,PT) WHERE INTEGER PTX,PT TCBE
BEGIN
COMMENT THIS PROCEDURE SEARCHES THE ACTIVE DICTIONARY FILE
TO FIND THE WORD POINTED TO BY PTX.  IF THE WORD IS
FOUND IN THE FILE, THEN THE DATA FOR THAT WORD IS TRANSMITTED
TO ARRAY P $,
          INTEGER M,DCTPT,BKPT,PN,WN,P1,COUNT,PTP,PTXX,PTDF,I $,
          PROCEDURE MOVE1 $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          P=W(PT+5) $,  ... GET WORD LENGTH//
          BKPT=W(PT+12) $,
          DCTPT=W(PT+13) $,  ... GET STARTING LOC OF FILE//
          PTP=W(PT+14)+W(PT+11) $,
          PA=W(BKPT+2*M) $,
COMMENT GET NUMBER AND LENGTH OF LAST FILE IN SERIES $,
          WA=W(BKPT+2*M+1) $,
          P1=P-1 $,
          COUNT=0 $,
START     I=0 $,
          IF W(PT+10) EQL PA THEN GOTO CHECK $,
COMMENT CHECK WHETHER FILE IS LAST IN SERIES $,
          IF COUNT GRT 862-P THEN GOTO RETURN $,
LOOP1     IF W(PT+I) NEQ W(DCTPT+COUNT+I) THEN GOTO CONT $,
COMMENT CHECK FOR A WORD MATCH $,
          I=I+1 $,
          IF I LES P1 THEN GOTO LOOP1 $,
          PTDF=DCTPT+COUNT+P+1 $,
          MOVE(PT,PTDF) $,  ... TRANSMIT DATA TO ARRAY P//
          GOTO RETURN $,
CONT      COUNT=COUNT+M+2 $,
          GOTO START $,
CHECK     IF COUNT LES WA THEN GOTO LOOP1 $,
COMMENT IF WORD NOT FOUND IN FILE THEN PUT -1 IN P ARRAY $,
          W(PTP)=-1 $,
END $,
DEFINE PROCEDURE CF2(ACT,PT) WHERE INTEGER ACT,PT TCBE
BEGIN
COMMENT THIS PROCEDURES OBTAINS THE SPECIFIED DICTIONARY FILE.
ACT MAY BE EITHER OPEN OR CLOSE.
          INTEGER BKPT,N,M,OPN,CLOS,N1,N2,NM1,NM2,RW,EOF,EOFCT $,
          INTEGER ARRAY OF(864) $,
          PROCEDURE OPEN,RDWAIT,CLOSE $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          PRESET BEGIN
            OPN  .BCD. /  OPEN/ $,
            CLOS .BCD. / CLOSE/ $,
            NM1=3733334500CC $,
COMMENT PRESET FILE NAME TO ...A00 ...P00 $,
            NM2=3333344000CC $,
            RW  .BCD. /R/ $,
          END $,
          W(PT+13)=LOC OF $,
          BKPT=W(PT+12) $,
          IF ACT EQL OPN THEN BEGIN
            M=W(PT+8) $,
COMMENT GET LENGTH AND NUMBER OF THE SPECIFIED FILE $,
            N=W(PT+10) $,
            IF N GEC 10 THEN N=(N/10)*66C+W(PT+10) $,
COMMENT GENERATE ECD NAME FOR FILE $,
            N1=NM1+N $,
            N2=NM2+M $,
            OPEN(RW,N1,N2,0,2) $,
            RDWAIT(N1,N2,0,OF TO 864,EOF,EOFCT) $,
          END $,
          IF ACT EQL CLOS THEN BEGIN
            CLOSE(N1,N2) $,
            W(PT+10)=W(PT+10)+1 $,
COMMENT INCREMENT FILE NUMBER FOR NEXT CALL $,
          END $,
END $,
END FIN.
```

```
COMMENT INITIALIZE THE COUNTING AND STATISTICS KEEPING VARIABLES FOR
THE PRE-INDEXING METHODS.  THE VARIABLES KEEP TRACT OF COMPLETENESS
AND RELEVANCE FOR ALL METHODS AND PARAMETER SETS.  THE PARAMETER SETS
ARE SUBSCRIPTED WITHIN EACH METHOD. $,
H=0.0 $,  ... NUMBER WORDS USED BY HUMAN INDEXERS//
C1(0)=C1(1)=C1(2)=0.0 $,
C2(0)=C2(1)=C2(2)=0.0 $,  ... NUMBER WORDS IN PRE-INDEX USED BY INDEXERS//
C3(0)=C3(1)=C3(2)=0.0 $,
R1(0)=R1(1)=R1(2)=0.0 $,
R2(0)=R2(1)=R2(2)=0.0 $,  ... SAME AS C1,2,3//
R3(0)=R3(1)=R3(2)=0.0 $,
RN1(0)=RN1(1)=RN1(2)=0.0 $,
RN2(0)=RN2(1)=RN2(2)=0.0 $,  ... NUMBER OF WORDS IN PRE-INDEX//
RN3(0)=RN3(1)=RN3(2)=0.0 $,
COMMENT SET SWITCHING INDEX AND INITIALIZE POINTERS $,
M=1 $,
START$ PTC=PTC1 $, PTP=PTP1 $, PTI=PTI1 $,
PTT=PTT1 $,
GOTO SW(M) $,
TITLE$ M=0 $, TN=0.0 $,
COMMENT FOR ALL METHODS THIS SECTION OF THE PROGRAM INCLUDES TITLE
WORDS IN THE PRE-INDEX AND MAKES APPROPRIATE EVALUATIONS $,
LOOPT$ PTT=PTT+P $,  ... GET NEXT TITLE WORD//
P=WLNGTH(PTT) $, ... FIND LENGTH OF WORD//
IF PTT GEQ W(PT)+W(PT+3) THEN BEGIN
COMMENT WHEN TITLE IS EXHAUSTED THEN EVALUATE THIS SECTION OF
PRE-INDEX AND RESET POINTERS TO ARRAYS P, PI, D, TO THE START OF THE
ABSTRACT SECTIONS OF EACH $,
RN1(0)=RN1(1)=RN1(2)=RN2(0)=RN2(1)=RN2(2)=RN3(0)=RN3(1)=RN3(2)=TN $,
PTC1=PTC $, PTP1=PTP $, PTI1=PTI $,
PTT1=PTT $,
GOTO NEXT $,
END $,
W(PTI)=777C $,  ... SET BIT FLAGS FOR ALL METHODS AND PARAMETER SETS//
IF W(PTC) GRT 0 THEN BEGIN
COMMENT UPDATE STATISTICAL DATA FOR ALL METHODS $,
J=0 $,
H=H+1.0 $
CIRCLE$ R1(J)=R1(J)+100.0 $, C1(J)=C1(J)+100.0 $,
R2(J)=R2(J)+100.C $, C2(J)=C2(J)+100.C $,
R3(J)=R3(J)+100.C $, C3(J)=C3(J)+100.C $,
J=J+1 $,
IF J LES 3 THEN GOTO CIRCLE $,
END $,
COMMENT THEN GET NEXT WORD OF TITLE $,
TN=TN+1.0 $,
PTC=PTC+1 $, PTP=PTP+1 $, PTI=PTI+1 $,
GOTO LOOPT $,
WLOOP$ M=0 $, XL=W(PT+3) $,
COMMENT START PRE-INDEXING BY METHODS I AND II $,
WLOOP $ IF W(PTP) EQL 0 THEN GOTO NEXT $,
COMMENT WHEN END OF ABSTRACT IS REACHED THEN GOTO NEXT METHOD $,
PTT=PTT+P $,  ... POINT TO NEXT WORD//
XL=XL-M $,
P=WLNGTH(PTT) $,
IF W(PTC) GRT 0 THEN H=H+1.C $,  ... IF INDEXERS USED WORD THEN UP COUNT//
IF W(PTP) EQL -1 THEN BEGIN
COMMENT IF THE WORD HAS NOT BEEN USED BEFORE, THEN INITIALIZE A CALL TO
WCHECK TO SEE IF THE WORD IS USED ELSEWHERE IN THE TA ARRAY. $,
W(PT+1)=W(PT)+XL .LS. 18 $,
W(PT+1)=PTT+M .LS. 18 $,
```

```
****************************************************************************************
     H5806        CMFL01          XRO        ALGOL FOR    PJDC6         4976        051960
BEGIN
DEFINE PROCEDURE EVAL(PT,B) WHERE INTEGER PT,B TOBE BEGIN
COMMENT THIS PROCEDURE ACTUALLY GENERATES PRE-INDEXES BY EACH OF
THE THREE METHODS.  BIT FLAGS ARE SET IN ARRAY PI TO
INDICATE WHICH WORDS ARE SELECTED FOR A PRE-INDEX.  THE PROCEDURE OPERATES
ON THE ARRAY P, WHICH CONTAINS THE DATA THAT HAS BEEN RETRIEVED
FROM THE DICTIONARY. THE ARGUMENT B POINTS TO AN ARRAY OF PARAMETERS WHICH
ARE USED IN THE IMPLEMENTATION OF THE METHODS. $,
INTEGER I,J,K,L,L2,M,R,F,PF,PR1,PR2,FLAG,BIT $,
INTEGER PTC1,NUMEND,XL,PTT1,PTP1,PTI1 $,
INTEGER PTP,PTX,PTD,PTI,PTT,PTB,PTB4,AST,ASTR,WRC $,
INTEGER ARRAY BLANK(8) $,
INTEGER PROCEDURE WLNGTH,WCHECK $,
PROCEDURE TYPASC,FRT,CARET $,
REAL TN $,
REAL ARRAY C1(3),C2(3),C3(3),R1(3),R2(3),R3(3) $,
REAL ARRAY RN1(3),RN2(3),RN3(3) $,
INTEGER COMPONENT W $,
W $=$ 0 $,
DEFINE PROCEDURE CX TOBE BEGIN
COMMENT THIS ROUTINE IS USED IN METHOD III TO ACT ON SPECIFIC
WORDS ONCE THE USE OF THE WORD HAS BEEN DECIDED $,
IF FLAG EQL 1 THEN BEGIN
COMMENT IF THE WORD IS INCLUDED IN A PRE-INDEX, THEN SET THE
APPROPRIATE FLAG BIT IN PI.  INCREMENT THE NUMBER OF WORDS INCLUDED
IN THE PRE-INDEX $,
W(PTI)=W(PTI) .V. (1 .LS. L2) $,
RN3(I)=RN3(I)+1.C $,
IF W(PTD) GRT 0 THEN BEGIN
COMMENT IF WORD HAS BEEN USED IN HUMAN SUBJECT INDEX THEN INCREMENT
COMPLETENESS AND RELEVANCE COUNTS FOR PRE-INDEX. $,
C3(I)=C3(I)+100.C $, R3(I)=R3(I)+100.C $,
END $,
END $,
PTI=PTI+1 $, PTD=PTD+1 $,
END $,
SWITCH SW=TITLE,PARSE,WLOOP,FINISH $,
COMMENT SET POINTERS TO INITIAL LOCATIONS OF THE VARIOUS ARRAYS.
PTC POINTS TO ARRAY CONTAINING ANALYSIS OF HUMAN SUBJECT INDEX.
PTI CONTAINS THE FLAG BITS TO MARK WORDS SELECTED FOR THE PRE-INDEXES.
PTP CONTAINS THE RESULTS OF THE DICTIONARY LOOKUPS FOR WORDS OF THE
TITLE AND ABSTRACT. $,
PTC1=W(PT+2)+1 $, PTP1=W(PT+14)+1 $, PTI1=W(PT+15)+1 $,
PTT1=W(PT) $,
PRESET BEGIN
ASTR=0400520400400C $,
BLANK(0)=040040C4C040C $,
BLANK(1)=04C04C04C040C $,
BLANK(2)=0400400C4C040C $,
BLANK(3)=0400400C4C040C $,
BLANK(4)=04004C04C040C $,
BLANK(5)=04C04CC4C040C $,
BLANK(6)=040040040C040C $,
BLANK(7)=04004C04C040C $,
END $,
AST=LOC ASTR $,
AST=AST+4C18 $,
PTP=LOC BLANK $,
NUMEND=W(PT+11) $,  ... GET NUMBER OF WORDS IN TA ARRAY//
```

```
IF WCHECK(IPT) EQL 0 THEN GOTO SETP $,
COMMENT IF THE WORD WAS BEEN USED ELSEWHERE THEN INCLUDE THE WORD IN
THE PRE-INDEX FOR BOTH METHODS I AND II. $,
I=C $,
ALLOC$ RN1[I]=RN1[I]+1.0 $, RN2[I]=RN2[I]+1.C $,
COMMENT MAKE THE STATISTICAL MEASUREMENT COUNTS $,
IF W[I+7D] EQL 0 THEN GOTO NB $,
R1[I]=R1[I]+100.C $, C1[I]=C1[I]+100.C $,
R2[I]=R2[I]+100.C $, C2[I]=C2[I]+100.C $,
NB$ I=I+1 $,
IF I LES 5 THEN GOTO BLOOP $,
W[PT]=W[PT] .V. 77C $, GOTO NWRD $,  ... SET FLAG BITS, GET NXT WORD//
ENC $,
COMMENT GET DICTIONARY DATA FROM ARRAY P $,
SETP$ P=P+(PTP) .A. 37C $, R=(W[PTP] .RS. 5) .A. 37C $,
COMMENT START WITH FIRST PARAMETER SET $,
I=0 $,
METH$ L=P+I $,
COMMENT COMPARE DATA WITH THRESHOLD PARAMETERS $,
IF R LEQ W[8+I] THEN GOTO METH2 $,
 W[PT]=W[PT] .V. (1 .LS. I) $,
RN1[I]=RN1[I]+1.C $,
IF W[7D] EQL 0 THEN GOTO METH2 $,
R1[I]=R1[I]+10H.C $, C1[I]=C1[I]+100.C $,
METH2$ IF W LEQ W[8+I+1] THEN GOTO NPARA $,
COMMENT COMPARE DATA WITH METHOD II THRESHOLD PARAMETERS $,
IF R .NOT. W[8+I+6] THEN GOTO NPARA $,
 W[PT]=W[PT] .V. (8 .LS. I) $,
RN2[I]=RN2[I]+1.C $,
IF W[7D] EQL 0 THEN GOTO NPARA $,
R2[I]=R2[I]+100.C $, C2[I]=C2[I]+100.C $,
COMMENT GET NEXT SET OF PARAMETERS $,
NPARA$ I=I+1 $, IF I LES 3 THEN GOTO METH1 $,
NWRD$ PTP=PTP+1 $, PTI=PTI+1 $, PTD=PID+1 $,
COMMENT C.U.T NOT DEC? $,
GOTO ALLOC2 $,
PARSE$ I=0 $,
COMMENT START METHOD III $,
PARSE.$ L=P+I $, L2=6+I $,
PTT=PTTL $,
BI=W[L+3] $, P=C $,
PTC=PTP1 $, PTP=PTP1 $, PTI=PTI1 $,
COMMENT NEW SET PARAMETERS TO BE USED $,
PR=W[L+1] $,
PR1=W[8+L+I] $, PR2=W[8+L+1+C] $,
WRC=0 $, FLAG=0 $,
ALLOC$ IF W(PTP) EQL 0 THEN GOTO WEAK $,
COMMENT IF THE END OF THE ATTRACT IS REACHED, THEN CONSIDER
THAT THE PACING IS A WEAK WORD $,
BL=BL+M $,
PTI=PTI+M $,
P=W[M.INC(PTI)] $,
IF W(PTP) EQL -1 THEN BEGIN
COMMENT IF THE WORD IS NEW THEN SET UP A CALL TO WCHECK TO SEE
IF THE WORD APPEARS ELSEWHERE IN TA $,
W[PT+1]=W[PTI+1]EL .LS. (8) $,
W[PT+1]=PTT+1P .LS. 18) $,
IF WCHECK(IPT) GRT 0 THEN GOTO STRONG ELSE GOTO MIDDLE $,
END $,
COMMENT GET THE DICTIONARY DATA FOR THE WORD $,
P=W[P(P) .A. 37C $, R=(W[PTP] .RS. 5) .A. 37C $,
COMMENT DETERMINE WHICH REGION OF USAGE RATE-FREQUENCY THE WORD
IS IN AND BRANCH ACCORDINGLY $,
IF P GRT P5 THEN GOTO WEAK $,
IF R LEQ PR1 THEN GOTO WEAK $,
IF R LEQ PR2 THEN GOTO MIDDLE $,
STRONG$ FLAG=1 $, GOTO UPDATE $,
MIDDL$ IF FLAG EQL 1 THEN GOTO UPDATE $,
COMMENT IF THE MIDDLE WORD IS NEXT TO A STRONG WORD (FLAG=1) THEN
INCLUDE IT IN THE PRE-INDEX. OTHERWISE POSTPONE THE DECISION. $,
WRL=W(D+1 $, GOTC NK $,
WEAK$ FLAG=0 $, GOTO UPDATE $,
UPDAT.$ IF WRD EQL 0 THEN BEGIN
COMMENT CHECK TO SEE IF ANY MIDDLE WORDS AWAIT DECISION $,
IF W(PTP) EQL 0 THEN GOTO NPARA2 $,
QX() $,
GOTC NK $,
END $,
QX() $, ... DECIDE A WAITING MIDDLE WORD//
WRC=W(D-1 $,  ... REDUCE QUEUE TO WAITING MIDDLE WORDS//
GOTC UPDATE $,
NK$ PTP=PTP+1 $,
GOTO PLOOP $,
NPARA2$ I=I+1 $,
IF I LES 3 THEN GOTO PARSE2 $,
GOTC NEXT $,
COMMENT INCREMENT THE SWITCHING INDEX TO PROCEDE TO THE NEXT SECTION
OF THE PROGRAM $,
NEXT$ K=K+1 $, GOTO START $,
FINISH$ PTC=W[PT+2]+1 $, PTP=W[PT+14]+1 $, PTI=W[PT+15]+1 $,
COMMENT TYPE OUT THE FINAL RESULTS $,
PTT=W[PT  $,
IF W[PT+16] EQL C THEN GOTO FINAL $,
COMMENT IF A FULL PRINTOUT IS NOT REQUESTED THEN BYPASS THE FULL
PRINTOUT AND GOTC THE SUMMARY PRINTOUT $
M=0 $, PTR=PTB+4C18 $,
WTYPE$ IF W(PTP) EQL 0 THEN GOTC FINAL $,
COMMENT WHEN THE END OF TA IS REACHED THEN GO PRINT OUT THE SUMMARY $,
PTI=PTT+P $, R=WLAGTH(PTT) $,  ... GET WORD AND ITS LENGTH//
COMMENT MAKE AN INTREX POINTER TO THE WORD AND TYPE IT OUT $,
PTR=PTT+P+4C18 $, TYPASC(PTR) $,
PTR=PTB+(9-P)+4C18 $, TYPASC(PTR) $,  ... PAD WITH BLANKS//
IF W(PTD) GRT 0 THEN TYPASC(AST) ELSE TYPASC(PTB4) $,
L=0 $,
TYP$ TYPASC(PTB4) $,
BI=(W(PTT) .RS. L) .A. 7C $,
COMMENT PRINT MARKER ASTERISK FOR EACH METHOD PARAMETER SET IN WHICH
THE WORD IS USED IN A PRE-INDEX $,
IF BI .A. 1 GRT C THEN TYPASC(AST) ELSE TYPASC(PTR4) $,
IF BI .A. 2 GRT C THEN TYPASC(AST) ELSE TYPASC(PTB4) $,
IF BI .A. 4 GRT C THEN TYPASC(AST) ELSE TYPASC(PTB4) $,
L=L+3 $, IF L LES 9 THEN GOTC TYP $,
PTI=PTI+1 $, PTC=PTD+1 $, PTP=PTP+1 $,
CAMET(1) $,
GOTO WTYPE $,
FINAL$ FOR J=0,1,2 DO BEGIN
COMMENT CALCULATE AND PRINT COMPLETENESS AND RELEVANCE
FOR EACH METHOD AND PARAMETER SET $,
IF N .NOT. 0.C THEN BEGIN
C1[J]=C1[J]/N $, C2[J]=C2[J]/N $, C3[J]=C2[J]/N $,
END $,
IF RN[J] GRT 0.C THEN R1[J]=R1[J]/RN1[J] $,
```

```
IF RA1[J] GRT 0.C THEN R2[J]=R2[J]/RN2[J] $,
IF RA3[J] GRT C.C THEN R3[J]=R3[J]/RN3[J] $,
ENC $,
PRINT FRA,RA1[0] RN1[1],RN1[2],RN2[0],RN2[1],RN2[2],RN3[C],RA3[1],RA3[2] $,
PRINT FN,% $,
PRINT FC,C1[0],C1[1],C1[2],C2[C],C2[1],C2[2],C3[C],C3[1],C3[2] $,
PRINT FR,R1[0],R1[1],R1[2],R2[C],R2[1],R2[2],R3[C];R3[1],R3[2] $,
PTP=W[PT+14] $, PTI=W[PT+15] $,
FOR J:1 STEP 1 WHILE J LEQ NUMEND DO BEGIN
W[PTP+J]=0 $, W[PTI+J]=0 $,
ENC $,
FN$ FORMAT(3HN =,27X,F4.0) $,
FRA$ FORMAT(/4HRA =,36X,3(3F4.C,4X)) $,
FC$ FORMAT(12HCOMPLETENESS,26X,3(3F6.C,4X)) $,
FR$ FORMAT(9HRELEVANCE,29X,3(3F4.0,4X)///) $,
ENC $,
ENC FINI
```

```
M5806      CMFL01          MR1        ALGOL FOR   M5806        4976      0515(8
BEGIN
DEFIN- PROCEDURE PEUR1(PT,PTDF) WHERE INTEGER PT,PTDF TCBE
BEGIN
COMMENT THIS PROCEDURE EXTRACTS USAGE RATE AND FREQUENCY
INFORMATION FROM A DICTIONARY FILE AND PLACES THE INFORMATION
IN THE ARRAY P $,
          INTEGER TAPP,AAPP,PTP,NDOC,TPT,APT,I,J $,
          INTEGER FR,RA,APP,TAPT $,
          INTEGER ARRAY TP(5),AP(5) $,
          INTEGER COMPONENT W $,
          W $=$ C $,
          PTP=W[PT+14]+W[PT+11] $,
COMMENT PTP POINTS TO THE LOCATION IN P WHERE THE
INFORMATION IS TO BE PLACED.  PTRS(14) CONTAINS THE STARTING LOCATION OF
ARRAY P AND PTRS(11) CONTAINS THE WORD NUMBER IN QUESTION. $,
          J=PTDF $,
COMMENT EXTRACT DATA ON USAGE AND APPEARANCES FROM THE FILE.
PTDF POINTS TO THE LAST OF THE THREE DATA WORDS $,
          TP[0]=W[J-2] .RS. 29 $,        AP[0]=W[J-1] .RS. 29 $,
          TP[1]=W[J-2] .A. 1777C $,          AP[1]=W[J-1] .A. 1777C $,
          TP[2]=W[J-2] .RS. 20) .A. 777C $, AP[2]=W[J-1] .RS. 20) .A. 777C $,
          TP[3]=W[J] .RS. 27 $,          AP[3]=(W[J] .RS. 9) .A. 777C $,
          TP[4]=W[J-2] .RS. 10) .A. 1777C $,     AP[4]=W[J-1] .RS. 10) .A. 1777C $,
          TAPP=W[J] .RS. 18) .A. 777C $,    AAPP=W[J] .A. 777C $,
          TPT=TP[C]+TP[1]+TP[2]+TP[3]+TP[4] $,
          APT=AP[C]+AP[1]+AP[2]+AP[3]+AP[4] $,
          NCOC=W[W[PT+12)+1] $,
          APP=TAPP+AAPP $,
          TAPT=TPT+APT $,
          FR=1+(3C*APP)/NCOC $,
COMMENT CALCULATE FREQUENCY ON A SCALE OF 1 TO 31 $,
          IF FR GRT 31 THEN FR=31 $,
          RA=0 $,
          IF APP EQL 0 THEN GOTO NEXT $,
          RA=1+(3C*TAPT)/APP $,   ... CALCULATE USAGE RATE//
          IF RA GRT 31 THEN RA=31 $,
NEXT$     W[PTP]=FR*RA*1C5 $,  ... TRANSMIT THE FREQUENCY AND USAGE RATE//
ENC $,
ENC FINI
```

BIBLIOGRAPHY

1. The American University, Machine Indexing: Problems and Progress (Papers presented at the Third Institute on Informatic Storage and Retrieval, February 13, 17, 1961), Washington, D. C., 1962, 3541.

2. Baxendale, P. B. "An Empirical Model for Computer Indexing," in Machine Indexing, American U., 1962, pp 207-218.

3. Bohnert, L. M., "New Role of Machines in Document Retrieval: Definitions and Scope," in Machine Indexing, American U., 1962, pp. 8-21.

4. Luhn, H. P. (ed), Automation and Scientific Communication, Short Papers, Part 1, American Documentation Institute, Washington, D. C., 1963, pp. 1-128.

5. Luhn, H. P., "Keyword-In-Context Index for Technical Literature (KWIC Index)" in Amer. Documentation 11, 1960, pp. 288-295.

6. Maron, M. E., "Automatic Indexing: An Experimental Inquiry," Report No. P-2180, 1 September 1960 (rev. 2 Feb. 1961), p.31.

7. McCormick, E. M., "Why Computers?" in Machine Indexing, American U., 1962, pp. 220-232.

8. Overhage, C., Harman, R. J. (ed). INTREX: Planning Conference, M.I.T. Press, Cambridge, Mass., 1965.

9. "Project INTREX: Semiannual Report," Massachusetts Institute of Technology, 15 Sept. 1967.

10. "Project INTREX: Semiannual Report," Massachusetts Institute of Technology, 15 March 1968.

11. Salton, G. (ed), "Information Storage and Retrieval," Scientific Rept. No. ISR-11, Department of Computer Science, Cornell University, Ithaca, New York, June, 1966.

12. Stevens, Mary E., "Automatic Indexing" A State-of-the-Art Report," U. S. Department of Commerce, National Bureau of Standards, NBS Monograph 91, 30 March 1965.